

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАСТОСУНКУ
З ГЕНЕТИЧНИМИ АЛГОРИТМАМИ ДЛЯ ПРОГНОЗУ ПОГОДИ
INFORMATION APP DEVELOPMENT WITH GENETIC
ALGORITHMS FOR WEATHER FORECASTING

Виконав: студент 4 курсу, групи КНТ-229

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

ЖИХАРЄВ Д.М.

(ПРИЗВИЩЕ та ініціали)

Керівник ЗЕЛЕНЬОВА І.Я.

(ПРИЗВИЩЕ та ініціали)

Рецензент КАЗУРОВА А.Є.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ.
Кафедра програмних засобів.
Ступінь вищої освіти бакалавр.
Спеціальність 122 Комп'ютерні науки.
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки.
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЖИХАРЄВ Данило Максимович

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка інформаційного застосунку з генетичними алгоритмами для прогнозу погоди. Information App Development with Genetic Algorithms for Weather Forecasting

керівник проєкту (роботи) к.т.н., доцент, ЗЕЛЕНЬОВА Ірина Яківна

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

затверджені наказом закладу вищої освіти від “04” квітня 2023 року № 87

2. Строк подання студентом проєкту (роботи) 01 червня 2023 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації.

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ЗЕЛЕНЬОВА І.Я., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “ 4 ” квітня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

_____ Данило ЖИХАРЄВ
(підпис) (Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Ірина ЗЕЛЕНЬОВА
(підпис) (Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 105 с., 9 табл., 38 рис., 4 дод., 15 джерел.

ГЕНЕТИЧНІ АЛГОРИТМИ, TRYSCACHE, CACHE, APACHE, JAVASCRIPT, ПРОГНОЗ ПОГОДИ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Об'єкт дослідження – процес розробки програмного забезпечення для прогнозу погоди.

Предмет дослідження – програмні засоби для прогнозу погоди.

Мета роботи – розробка програмного засобу, здатного прогнозувати погоду для певного регіону, побудованого на базі машинного навчання нейронних мереж генетичними алгоритмами.

Матеріали, методи та технічні засоби: програмний продукт створений на мові COS, в середовищі Cache: Tryscache. В якості сервера був використаний спеціально адаптований під дане середовище сервер Apache 2.2. Також в основі розробленого модуля використовувався JavaScript.

Для досягнення мети роботи були поставлені та виконані наступні задачі:

- аналіз аналогічних систем;
- аналіз архітектури та можливостей середовища Tryscache;
- навчання нейронної мережі генетичними алгоритмами;
- тестування можливостей розробленого продукту.

Результати. Створено інформаційний застосунок з генетичними алгоритмами для прогнозу погоди.

Галузь використання – перегляд актуального прогнозу погоди.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 105 pages, 15 tables, 38 figures, 4 appendixes, 15 sources.

GENETIC ALGORITHMS, TRYCACHE, CACHE, APACHE, JAVASCRIPT, WEATHER FORECAST, SOFTWARE.

The object of research is the process of developing software for weather forecasting.

The subject of research is software for weather forecasting.

The purpose of the work is to create a software tool capable of forecasting the weather for a certain region, built on the basis of machine learning of neural networks using genetic algorithms.

Materials, Methods and Techniques: The software product is written in COS language, among Cache: Trycache. An Apache 2.2 server specially adapted for this environment was used as a server. JavaScript was also used as the basis of the developed module.

To achieve the goal of the work, the following tasks were set and completed:

- analysis of similar systems;
- analysis of the architecture and capabilities of the Trycache environment;
- neural network training with genetic algorithms;
- testing the capabilities of the developed product.

The results. An informational application with genetic algorithms for weather forecasting has been created.

The field of use is viewing the current weather forecast.

ЗМІСТ

С.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Визначення поняття «інформаційна система»	10
1.2 Засоби передачі відповідних даних та повідомлень та засоби збереження інформації.....	11
1.3 Аналіз існуючих систем	11
1.4 Засоби аналізу, обробки і представлення інформації.....	13
1.5 Постановка завдання до роботи.....	14
2 ВИБІР ТА ОБГРУНТУВАННЯ СИСТЕМИ ЯКА ПРОЕКТУЄТЬСЯ.....	16
2.1 Сучасні моделі прогнозування	16
2.1.1 Чисельна модель прогнозування	16
2.1.2 Модель прогнозування центру European Centre for medium-range weather forecasts (ECMWF).....	17
2.1.3 Модель Weather Research and Forecasting (WRF)	18
2.1.4 Модель Навчання штучних нейронних мереж генетичними алгоритмами.....	19
2.2 Сучасні СКБД.....	24
2.2.1 СКБД MySQL	24
2.2.2 СКБД Oracle.....	25
2.2.3 СКБД Cache Intersystems	26

	7
2.4 Cache Server Page (CSP).....	27
2.5 JavaScript	28
2.6 HTML і CSS	30
3 РОЗРОБКА ОСНОВНИХ ПРОГРАМНИХ ЧАСТИН.....	34
3.1 ER діаграма програми.....	34
3.2 Алгоритм роботи програми.....	35
3.6 Структура середовища розробки.....	36
4 ТЕСТУВАННЯ ТА ДОСЛІДЖЕННЯ ВИГОТОВЛЕНОЇ ПРОГРАМИ....	43
4.1 Призначення й умови застосування програми.....	43
4.2 Характеристики програми.....	43
4.3 Звертання до програми	45
4.4 Повідомлення.....	46
4.5 Виконання програми.....	47
4.6 Порівняння інтелектуальних методів для прогнозування погоди	48
ВИСНОВКИ.....	51
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	52
ДОДАТОК А Програмний код компонентів сайту	54
ДОДАТОК Б Слайди презентації	98

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API – application programming interface (прикладний програмний інтерфейс);

CSS – content management system (система управління вмістом);

HTML – HyperText Markup Language (язык гипертекстовой разметки);

БД – база даних;

ІС – інформаційна система;

СКБД – система керування базами даних;

ШНМ – штучна нейронна мережа.

ВСТУП

На сьогоднішній день існує дуже багато різноманітних інформаційних систем. Вони бувають різних видів від інформаційних систем прогнозування погоди, до інформаційних систем прогнозування курсу валют. Інформаційна система – це ресурс, де користувач може отримати потрібну йому інформацію у достатньому обсязі. Але це не означає, що на цьому функціональність інформаційної системи закінчується, так як перш за все система – це цілий комплекс компонентів пов’язаних між собою і в деяких випадках здатний до самонавчання. Саме тому було необхідно було створити ресурс, який зміг би не тільки аналізувати та обробляти інформацію, але ще і самонавчатися та розвиватися самостійно [1].

Розроблений ресурс має виконувати такі функції:

- за допомогою алгоритмів робити прогноз погоди;
- самостійно оцінювати ефективність алгоритмів;
- спираючись на оцінку алгоритмів скласти найдостовірніший прогноз погоди;
- надавати користувачам інформацію про погоду в межах певного регіону.

В інформаційних застосунках, призначенням яких є прогноз погоди засобами аналізу виступають електронно-обчислювальні машини, які перш за все отримують інформацію із зовнішніх джерел, аналізують її, обробляють, і після того представляють інформацію у виді передбаченому оператором [2].

Основна ціль розробленого інформаційного застосунку – реалізація новітнього способу прогнозування та надання інформації користувачу про спрогнозовані дані, у нашому випадку це прогноз погоди, для певної місцевості, що буде на наступний день. Інтерфейс має бути простим та зрозумілим [2].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Визначення поняття «інформаційна система»

Інформаційна система (англ. Information system, IC) — сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів [3].

Інформаційна система може існувати і без застосування комп'ютерної техніки — це питання економічної доцільності.

В будь-якій інформаційній системі управління вирішуються задачі трьох типів:

- задачі оцінки ситуації (деколи їх називають задачами розпізнавання образів);
- задачі перетворення опису ситуації (розрахункові задачі, задачі моделювання);
- задачі прийняття рішень (в тому числі і оптимізаційні).

Найдавнішими і найпоширенішими IC слід вважати бібліотеки. І, дійсно, здавна в бібліотеках збирають книжки (або їх аналоги), зберігають їх, дотримуючись певних правил, створюють каталоги різного призначення для полегшення доступу до книжкового фонду. Видаються спеціальні журнали та довідники, що інформують про нові надходження, ведеться облік видачі.

Інформаційні системи включають в себе: технічні засоби обробки даних, програмне забезпечення і відповідний персонал [3]. Чотири складові частини утворюють внутрішню інформаційну основу:

- засоби фіксації і збору інформації;
- засоби передачі відповідних даних та повідомлень;
- засоби збереження інформації;
- засоби аналізу, обробки і представлення інформації.

Тож коли ми розібралися що являє собою інформаційна система, розберемося із її складовими. Розглянемо вище вказані складові для нашої

інформаційного застосунку з генетичними алгоритмами для прогнозу погоди [3].

1.2 Засоби передачі відповідних даних та повідомлень та засоби збереження інформації

В інформаційних системах, призначенням яких є прогноз погоди, засобами передачі відповідних даних та повідомлень виступають спеціалізовані вебсервери, та інформація, отримана з метеостанції.

В інформаційних системах, призначенням яких є прогноз погоди та в багатьох інших інформаційних системах засобами для збереження інформації виступають бази даних (БД).

1.3 Аналіз існуючих систем

Сучасні інформаційні системи базуються на різноманітних методах прогнозування, наприклад European Centre for medium-range weather forecasts (ECMWF) використовує спеціальну модель на основі примітивних рівнянь (spectral primitive equation model) з полулагранжевою (semi-lagrangian) і напівнеявними (semi-implicit) тимчасовими схемами і складним тлумаченням фізичних процесів. В той час як Weather Research and Forecasting (WRF) має два динамічних ядра, систему асиміляції даних, а також програмну архітектуру, яка полегшує паралельні обчислення і розширює систему [4].

Якщо розглядати інтерфейс таких інформаційних систем, то можна побачити що він досить різноманітний. Наприклад <https://meteo.ua/ua>, зображений на рис. 1.1, надає інформацію про фактичні дані погоди, які знімаються з метеостанцій у різних містах [5].

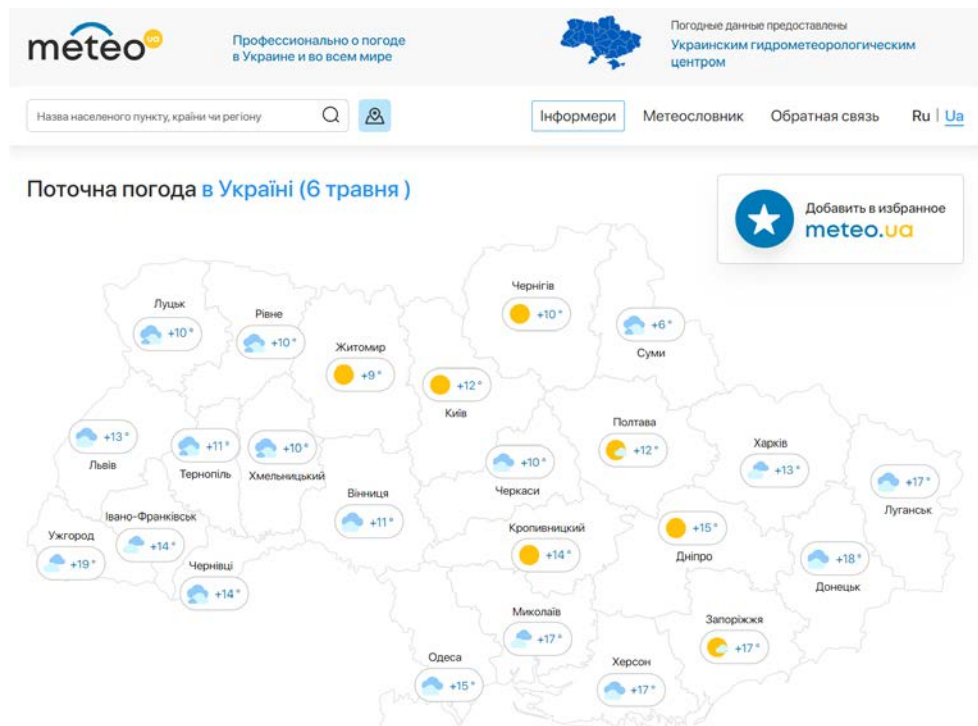


Рисунок 1.1 – Інформаційна система meteo.ua [5]

А ось інша, подібна до першої, інформаційна система <https://www.meteoprog.com/ua/weather/Zaporizhzhia/> представляє дані про спрогнозовану температуру рис. 1.2, [6].

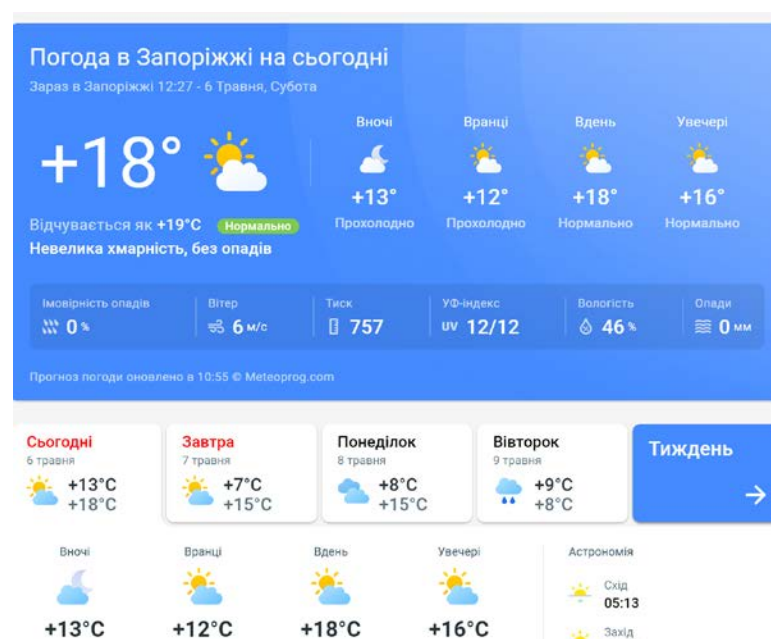


Рисунок 1.2 – Інформаційна система meteorprog [6]

А от популярна інформаційна система **Gismeteo** на рис. 1.3 надає лише необхідні звичайному користувачу дані у досить зрозумілій і зручній формі [7].

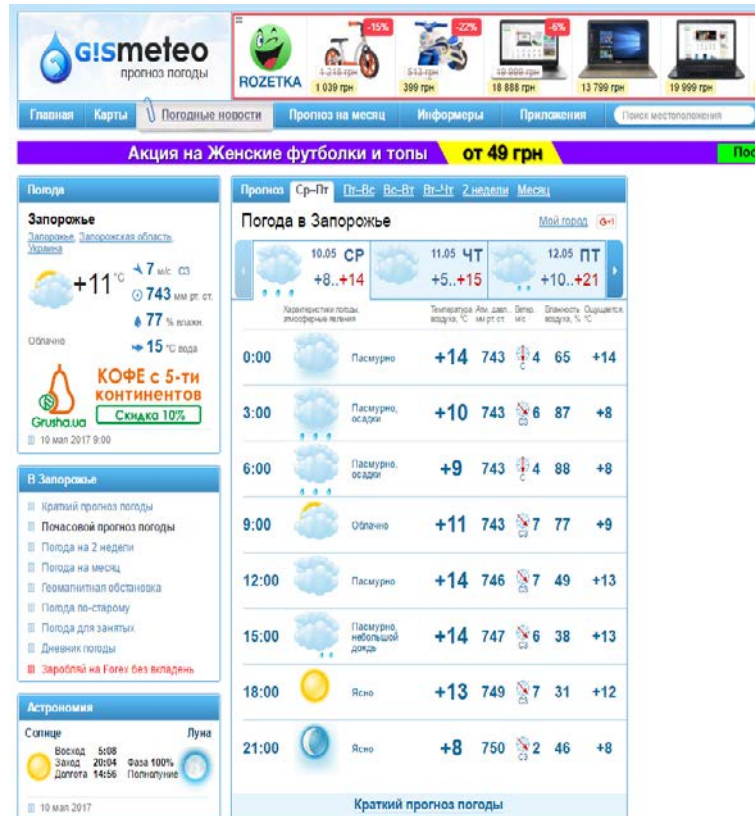


Рисунок 1.3 – Інформаційна система Gismeteo [7]

Отже ознайомившись з інтерфейсом існуючих інформаційних систем можна зробити висновок, що роль користувача у них зведена до мінімуму, а сам інтерфейс повинен бути простим та зрозумілим.

1.4 Засоби аналізу, обробки і представлення інформації

В інформаційних системах, призначенням яких є прогноз погоди засобами аналізу виступають електронно-обчислювальні машини, які перш за все отримують інформацію із зовнішніх джерел, аналізують її, обробляють, і після того представляють інформацію у виді передбаченому оператором.

Тепер, коли ми маємо загальне уявлення про проєктовану інформаційну систему, слід розглянути дуже важливу частину, а саме те, як наша інформаційна система буде взаємодіяти із користувачем.

Якщо розглядати конкретно нашу інформаційну систему «Прогноз погоди», то це має бути загальнодоступний інтернет-ресурс, тож очевидно що наша інформаційна система буде розроблятися як вебпродукт.

Перш за все необхідно знайти вебхостинг де буде розміщено наш вебпродукт. Обчислювальні машини вебхостингу мають повністю задовольняти вимогам нашої інформаційної системи і тільки після цього можна починати розроблення нашої інформаційної системи у виді вебпродукту.

Для того щоб наш вебпродукт користувався популярністю він повинен мати простий, зрозумілий і в той же час цікавий інтерфейс.

Тож, коли наша інформаційна система матиме фізичне місце(ЕОМ в сукупності із БД) її розташування, доступ до неї буде відкритий для великої кількості людей, у нашому випадку цей доступ буде здійснений за допомогою вираження нашої інформаційної системи через вебпродукт, і матиме свою аудиторію, і буде стабільно функціонувати лише тоді вона буде повноцінною.

1.5 Постановка завдання до роботи

Прогнозування погоди має особливо велике значення для агропромислових підприємств та звичайних людей. Для роботи було обрано дані котрі містять числові показники, що характеризують сезонні зміни погоди, БД постійно поповнюється поточним станом погоди (вітер, дощ, температура тощо).

Метою роботи - розробка програмного засобу, здатного прогнозувати погоду для певного регіону, побудованого на базі машинного навчання нейронних мереж генетичними алгоритмами.

Для досягнення цілі роботи були поставлені та виконані наступні задачі:

- проаналізувати аналогічні системи засновані на машинному навчанні нейронних мереж;
- вивчити архітектуру та можливості середовища Trycache;
- навчити нейронні мережі генетичними алгоритмами;
- продемонструвати можливості розробленого продукту на прикладі прогнозування погоди.

2 ВИБІР ТА ОБГРУНТУВАННЯ СИСТЕМИ ЯКА ПРОЕКТУЄТЬСЯ

2.1 Сучасні моделі прогнозування

Сьогодні існує багато інформаційних систем, які використовують різноманітні моделі прогнозування, тому перш за все розглянемо моделі, на яких побудовані такі системи, призначенням яких є прогнозування погоди, або конкретні приклади відомих систем прогнозування погоди.

2.1.1 Чисельна модель прогнозування

Чисельна модель прогнозування погоди - це комп'ютерна програма, побудована на основі фізичної системи рівнянь і складає на основі поточних даних метеорологічний прогноз. Ця модель може бути глобальною, що покриває всю Землю, або локальною, що покриває окрему ділянку планети.

В основі моделей лежать математичні рівняння, що описують аеро- і термодинамічні процеси в атмосфері і зв'язуючі такі параметри як щільність, швидкість, тиск і температуру. Ці рівняння є нелінійними і не мають точного рішення, тому для їх вирішення використовуються чисельні методи. Вихідні рівняння дискредитуються в часі і просторі і перетворюються в систему лінійних рівнянь, що зв'язує набори фізичних параметрів в обраних точках (вузлах обчислювальної сітки). Чим більше використовується точок для розрахунку, тим вище точність моделі, а й тим вище вимоги до обчислювальних потужностей ЕОМ.

В якості вихідних даних для моделей використовуються дані метеозондів, метеосупутників і наземних метеостанцій [8].

2.1.2 Модель прогнозування центру European Centre for medium-range weather forecasts (ECMWF)

Короткострокові моделі прогнозування зараз дуже складні. Найчастіше вони є комбінаціями глобальних і локальних моделей. Розглянемо роботу European Centre for medium-range weather forecasts (ECMWF).

На сьогоднішній день даний центр є світовим лідером в розробці та удосконаленні моделей прогнозування погоди. Перший прогноз в ECMWF був розрахований 1 серпня 1979 року [8].

ECMWF використовує спеціальну модель на основі примітивних рівнянь (spectral primitive equation model) з полулагранжевою (semi-lagrangian) і напівнеявними (semi-implicit) тимчасовими схемами і складним тлумаченням фізичних процесів. Розширення моделі 25 на 25 км має 91 рівень по вертикалі. Початкові умови для розрахунку готує чотирьох розмірна схема асиміляції (four-dimensional variational assimilation scheme), яка використовує дані з супутників в 12-годинному вікні (12-hour time window) [9].

ECMWF робить такі прогнози:

- прогноз погоди на 10 днів вперед, виконується двічі на день (модель з розширенням 25 на 25 км, 91 рівень по вертикалі);
- прогноз на підставі множинної системи прогнозування (ensemble prediction system, EPS) виконується двічі на день (модель 50 на 50 км, 62 рівня по вертикалі);
- прогноз на місяць вперед, виконується раз на тиждень (модель 78 на 78 км, 62 рівня по вертикалі);
- сезонний прогноз більш 6 місяців вперед, виконується раз на місяць (модель 125 на 125 км, 40 рівнів по вертикалі).

Для обчислень ECMWF використовує складні комп'ютерні системи. У серці системі знаходиться IBM High Performance Computing Facility (HPCF). HPCF включає два однакових кластера p690 +. Кожен кластер складається з 68 серверів, кожен з яких має 32 CPU з частотою 1.9 GHz. Пікова продуктивність складає 16.5 teraflops (трильйон операцій з плаваючою крапкою в секунду) на кожен кластер, тобто система може виконувати 33 трильйона операцій в секунду [9].

Для порівняння: HPCF на десять порядків більш продуктивна ніж ENIAC.

2.1.3 Модель Weather Research and Forecasting (WRF)

«The Weather Research and Forecasting (WRF)» модель середньомасштабної системи чисельного прогнозування погоди нового покоління призначена як для атмосферних досліджень так і для потреб оперативного прогнозування. Вона має два динамічних ядра, систему асиміляції даних, а також програмну архітектуру, яка полегшує паралельні обчислення і розширює систему. Модель пропонує широкий спектр метеорологічних додатків в масштабах від десятків метрів до декількох тисяч кілометрів. Зусилля по розробці WRF почалися в другій половині 1990-х років і була спільними партнерами із National Center for Atmospheric Research (NCAR), National Oceanic and Atmospheric Administration (в його складі значилася як National Centers for Environmental Prediction (NCEP) а потім як Forecast Systems Laboratory (FSL)), Air Force Weather Agency (AFWA), Naval Research Laboratory, the University of Oklahoma, і нарешті з Federal Aviation Administration (FAA) [9].

WRF може генерувати атмосферні моделювання з використанням реальних даних (спостереження, аналізи) або ідеалізовані умови. WRF пропонує оперативне прогнозування гнучкою і обчислювально-ефективною

платформою, забезпечуючи при цьому останні досягнення в галузі фізики, чисельних методів, і асиміляції даних наданих розробниками з наукового співтовариства. WRF на даний час значиться в оперативному використанні у гіперхолестеринемії, AFWA та інших центрах [9].

WRF має велике всесвітнє співтовариство користувачів (zareєстрованих більше 30000 в більш ніж 150 країнах), а також семінари та навчальні засідання, які проводяться щороку в NCAR. Система WRF містить два динамічних вичислювача, таких як ядра ARW (Advanced Research WRF) і ядро HMM (Nonhydrostatic Mesoscale Model). ARW був розроблений і підтримується лабораторією MMM, і сторінка його користувачів: WRF-ARW Users 'Page. Ядро HMM було розроблено National Centers for Environmental Prediction. В даний час використовується в системі HWRF (Hurricane WRF), для яких підтримка користувачів забезпечується за допомогою розвиваючого Testbed центру (див WRF для Hurricanes.) [9].

2.1.4 Модель Навчання штучних нейронних мереж генетичними алгоритмами

Генетичний алгоритм – це еволюційний алгоритм пошуку, що використовується для вирішення задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації параметрів з використанням механізмів, що нагадують біологічну еволюцію [10].

Особливістю генетичного алгоритму є акцент на використання оператора «схрещення», який виконує операцію рекомбінацію рішень-кандидатів, роль якої аналогічна ролі схрещення в живій природі. «Батьком-засновником» генетичних алгоритмів вважається Джон Голланд (англ. John Holland), книга якого «Адаптація в природних і штучних системах» (англ. *Adaptation in Natural and Artificial Systems*) є фундаментальною в цій сфері досліджень.

Задача кодується таким чином, щоб її вирішення могло бути представлено в вигляді масиву подібного до інформації складу хромосоми. Цей масив часто називають саме так «хромосома». Випадковим чином в масиві створюється деяка кількість початкових елементів «осіб», або початкова популяція. Особи оцінюються з використанням функції пристосування, в результаті якої кожній особі присвоюється певне значення пристосованості, яке визначає можливість виживання особи. Після цього з використанням отриманих значень пристосованості вибираються особи, допущені до схрещення (селекція). До осіб застосовується «генетичні оператори» (в більшості випадків це оператор схрещення (crossover) і оператор мутації (mutation)), створюючи таким чином наступне покоління осіб. Особи наступного покоління також оцінюються застосуванням генетичних операторів і виконується селекція і мутація. Так моделюється еволюційний процес, що продовжується декілька життєвих циклів (поколінь), поки не буде виконано критерій зупинки алгоритму. Таким критерієм може бути:

- знаходження глобального, або надоптимального вирішення;
- вичерпання числа поколінь, що відпущені на еволюцію;
- вичерпання часу, відпущеного на еволюцію.

Генетичні алгоритми можуть використати для пошуку рішень в дуже великих і важких просторах пошуку [10].

Між собою генетичні алгоритми поділяються на декілька груп, основними із яких є:

- група алгоритмів відбору, серед них: відбір ранжируванням, пороговий, пропорційний та турнірний відбори;
- група алгоритмів схрещування: арифметичний, одиничний, порівняльний та рівномірний алгоритми;
- група алгоритмів мутації: алгоритм простої, випадкової, точкової та мутації Гаусса.

Штучна нейронна мережа (ШНМ) — це математична модель, а також її програмна та апаратна реалізація, побудовані за принципом функціонування біологічних нейронних мереж — мереж нервових клітин живого організму. Це поняття виникло при вивченні процесів, які відбуваються в мозку, та при намаганні змодельовати ці процеси. Першою такою спробою були нейронні мережі У. Маккалока та У. Піттса. Після розробки алгоритмів навчання отримувані моделі стали використовуватися в практичних цілях: в задачах прогнозування, для розпізнавання образів, в задачах керування тощо [10].

ШНМ являють собою систему з'єднаних між собою простих обробників (штучних нейронів), які взаємодіють. Такі обробники зазвичай є доволі простими (особливо в порівнянні з процесорами, що застосовуються в персональних комп'ютерах). Кожен обробник подібної мережі має справу лише з сигналами, які він періодично отримує, і сигналами, які він періодично надсилає іншим обробникам. І тим не менш, будучи з'єднаними в достатньо велику мережу з керованою взаємодією, такі локально прості обробники разом здатні виконувати доволі складні завдання:

- з точки зору машинного навчання, нейронна мережа є окремим випадком методів розпізнавання образів, дискримінантного аналізу, методів кластерування тощо;
- з математичної точки зору, навчання нейронних мереж — це багатопараметрична задача нелінійної оптимізації;
- з точки зору кібернетики, нейронна мережа використовується в задачах адаптивного керування, і як алгоритми для робототехніки;
- з точки зору розвитку обчислювальної техніки та програмування, нейронна мережа — спосіб розв'язання задачі ефективного паралелізму.

Нейронні мережі не програмуються в звичайному розумінні цього слова, вони навчаються. Можливість навчання — одна з головних переваг нейронних мереж перед традиційними алгоритмами. Технічно, навчання полягає в знаходженні коефіцієнтів зв'язків між нейронами. В процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними й вихідними, а також здійснювати узагальнення. Це означає, що в

разі успішного навчання мережа зможе повернути правильний результат на підставі даних, які були відсутні в навчальній вибірці, а також неповних та/або «зашумлених», частково спотворених даних [10]. Порівняння методів наведено у табл.2.1.

Таблиця 2.1 – Порівняння алгоритмів

Метод	Переваги	Недоліки
1	2	3
Чисельна модель прогнозування	Легко розуміти та реалізовувати.	Не бере до уваги тенденції та сезонність ряду, тому не дуже точний для прогнозування довгострокових тенденцій або змін в сезонності.
	Швидко вираховує прогнози та використовує мало обчислювальних ресурсів.	Дуже чутливий до випадкових змін, тобто враховує вагу кожного спостереження однаково, що може призводити до неточних прогнозів при виникненні великих випадкових змін.
2 Модель прогнозування центру European Centre for medium-range weather forecasts	Легко розуміти та реалізовувати.	Не бере до уваги тенденції та сезонність ряду, тому не може бути використаний для прогнозування рядів з сезонністю або трендом.

Продовження таблиці 2.1

1□	2□	3□
□	Показує кращі результати в порівнянні з чисельною моделлю прогнозування в більш складних рядах.□	Не ефективний для рядів, де інтервали часу різні, наприклад, для даних з місячною та щоденною частотою.□
Модель Weather Research and Forecasting (WRF)□	Бере до уваги тенденції та сезонність ряду, що дозволяє більш точно прогнозувати дані.□	Не рекомендується для рядів зі складними тенденціями та сезонністю, так як цей метод не може обробляти більш складні ряди.□
	Легко налаштовується для різних характеристик ряду, що дозволяє працювати з різними типами даних.□	Чутливий до випадкових змін в даних, що може призвести до неприродних змін у прогнозах.□
Модель Навчання штучних нейронних мереж генетичними алгоритмами-я□	Бере до уваги тенденції та сезонність ряду, що дозволяє більш точно прогнозувати дані.□	Вимагає великої кількості даних для належного підбору параметрів.□
	Легко налаштовується та підтримує автоматичне оновлення параметрів, що зменшує витрати часу та зусиль.□	Не може виявляти зміну в режимі роботи, якщо тенденція та сезонність ряду змінюються швидко.□

2.2 Сучасні СКБД

Тепер, коли ми маємо уяву про те як здійснюється прогнозування інформації, нам необхідно розглянути ще особливості існуючих систем керування базами даних(СКБД).

2.2.1 СКБД MySQL

MySQL — вільна система керування реляційними базами даних.

MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних вебсторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.[11]

MySQL — компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання.

MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому.

Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;

- наявність простої і ефективної системи безпеки [11].

2.2.2 СКБД Oracle

Oracle Database – це об'єктно-реляційна система підтримує деякі технології, які реалізують об'єктно-орієнтований підхід, тобто забезпечують управління створення та використання баз даних [12].

Ключові можливості Oracle Database:

- Real Application Cluster (RAC) забезпечує роботу одного примірника бази даних на декількох вузлах grid, дозволяючи управляти навантаженням і гнучко масштабувати систему в разі потреби;
- Automatic Storage Management (ASM) дозволяє автоматично розподіляти дані між наявними ресурсами систем зберігання даних, що підвищує відмовостійкість системи і знижує загальну вартість володіння (TCO);
- продуктивність. Oracle Database дозволяє автоматично управляти рівнями сервісу і тиражувати еталонні конфігурації в рамках всієї мережі.
- прості засоби розробки. Новий інструмент розробки додатків HTML DB дозволить простим користувачам створювати ефективні програми для роботи з базами даних в короткі терміни;
- самоврядування. Спеціальні механізми Oracle Database дозволяють самостійно перерозподіляти навантаження на систему, оптимізувати і коригувати SQL-запити, виявляти і прогнозувати помилки.
- великі бази даних. Тепер максимальний розмір екземпляра бази даних Oracle може досягати 8 екзабайт;
- недорогі серверні системи. Oracle Database може використовувати недорогі однопроцесорні комп'ютери або модульні системи з "серверів-лез".

У новій версії бази даних реалізована підтримка перенесення табличних просторів, система управління потоками даних Oracle Streams та

модель розподілених SQL-запитів. Для перенесення існуючих баз даних в середу Grid в них не буде потрібно вносити змін, що дозволяє швидко почати використовувати всі переваги Oracle Database [12].

2.2.3 СКБД Cache Intersystems

Відкритість і гнучкість - головні характеристики, що відрізняють процес розробки продуктів на базі Caché. Caché надає потужну і просту у використанні інтегровану середу розробки для швидкого створення мобільних і вебдодатків за допомогою цілого ряду мов програмування [12].

Caché ObjectScript - потужний і простий у вивченні об'єктно-орієнтована мова для створення програм і об'єктних методів. Це один з трьох повністю сумісних між собою мов сценаріїв, які можуть використовуватися у віртуальній машині Caché Virtual Machine. Інші дві мови - Caché Basic і MVBasic. Ці мови дозволяють безпосередньо обробляти багатовимірні структури даних Caché і тому часто використовуються в системах, де продуктивність є пріоритетом.

Мови сценаріїв, підтримувані Caché, зрозуміло, також можна використовувати в програмуванні на стороні клієнта, проте існують і інші варіанти. Завдяки об'єктної технології, що є основою Caché, ця система підтримує декілька поширених мов програмування, так що ви зможете працювати зі звичними технологіями та інструментами. Технологія Caché eXtreme забезпечує «легковаге зв'язування» для об'єктів на мові Java або C++. Це означає, що об'єкти можна дуже швидко безпосередньо вставляти в багатовимірну базу даних Caché [12].

У сучасному світі, який немислимий без Інтернету, XML і вебслужби часто застосовуються для обміну даними та спільного використання функціональних можливостей в додатках. Caché спрощує експорт даних у вигляді XML, синтаксичний розбір і збереження даних з вхідних XML-

файлів, а також перетворення будь-якого класу Caché в вебслужбу. Caché, крім того, підтримує декілька стандартів для захисту вебслужб.

Вебінтерфейси додатків сьогодні стали нормою, і багато додатків адаптують для використання на мобільних пристроях або створюють такими спочатку [12].

Технологія InterSystems Zen дозволяє створювати потужний вебінтерфейс для нових або вже використовуваних додатків.

Завдяки розширенню технології Zen - Zen Mojo - ви можете швидко розробити наповнені даними веб і мобільні додатки. Крім того, можна використовувати вебслужби, REST, JSON і інші стандарти, а також обрані вами інструменти для створення мобільних додатків, які будуть відрізнятися феноменальною швидкістю і масштабістю Caché.

Caché має сертифікат відповідності Common Criteria EAL 3. Це не просто безпечне середовище - це зручний засіб створення компонентів захисту для передових продуктів. Caché дозволяє виконувати перевірку автентичності, авторизацію, аудит і шифрування баз даних практично без шкоди для продуктивності програмних продуктів [12].

2.4 Cache Server Page (CSP)

Caché Server Pages (CSP) розроблювався як архітектура і набір інструментів для створення інтерактивних додатків CSP. Технологія CSP дозволяє створювати і розгортати високопродуктивні, масштабовані вебдодатки. CSP дозволяє динамічно створювати вебсторінки, як правило, з використанням даних з бази даних Caché. «Динамічно» означає, що одна і та ж сторінка може представляти різний зміст кожного разу, коли сервер відправляє запит до нещодавно змінених джерел даних [13].

CSP є універсальним інструментом. Це можуть підтвердити дані пов'язані із сучасними Медіа, що змінюються кожної хвилини та вебспільнота з тисячами активних користувачів.

CSP надає можливість персоналізувати сторінки спираючись на інформацію користувача, що зберігається в базі даних Cache.

CSP здатна відображати HTML, XML, зображення або інші виконавчі або текстові дані [13].

CSP забезпечує високу продуктивність, так як тісно пов'язаний з базою даних Cache.

CSP добре підходить для додатків до баз даних. Крім забезпечення швидкого доступу до вбудованої бази даних Cache, CSP надає ряд функцій, необхідних для вебдодатків на основі баз даних, включаючи:

- управління сеансами;
- сторінку аутентифікації;
- можливість виконання інтерактивних операцій з базами даних та вебсторінками;
- CSP підтримує два стилі веброзробки;
- для розробки додатків з використанням класів, CSP забезпечує структуру об'єкта;
- для розробки додатків з використанням HTML-файлів, CSP забезпечує HTML на основі мови розмітки, що дозволяє включити об'єкти із серверних сценаріїв в середину вебсторінок. Програміст може об'єднати ці два методи в додатку для забезпечення максимальної гнучкості.

2.5 JavaScript

JavaScript (аббр. JS) – прототипно-орієнтована сценарна мову програмування. Є реалізацією мови ECMAScript [14].

JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить в браузерах як мова сценаріїв для додання інтерактивності вебсторінок.

Основні архітектурні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу.

На JavaScript вплинули багато мов, при розробці була мета зробити мову схожою на Java, але при цьому легку для використання непрограмістами. Мовою JavaScript не володіє будь-яка компанія або організація, що відрізняє її від ряду мов програмування, використовуваних в веброботі.

Назва «JavaScript» є зареєстрованим товарним знаком компанії Oracle Corporation [14].

JavaScript є об'єктно-орієнтованою мовою, але використовуване в мові прототипування обумовлює відмінності в роботі з об'єктами в порівнянні з традиційними клас-орієнтованими мовами. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам – функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання – що додає мові додаткову гнучкість [14].

Незважаючи на схожий з Сі синтаксис, JavaScript в порівнянні з мовою Сі має корінні відмінності:

- об'єкти, з можливістю інтроспекції;
- функції як об'єкти першого класу;
- автоматичне приведення типів;
- автоматичне прибирання сміття;
- анонімні функції.

У мові відсутні такі корисні речі, як:

- модульна система: JavaScript не надає можливості управляти залежностями і ізоляцією областей видимості;

- стандартна бібліотека: зокрема, відсутній інтерфейс програмування додатків в роботі з файловою системою, управління потоками введення-виведення, базових типів для бінарних даних;
- стандартні інтерфейси до вебсерверів і баз даних;
- система управління пакетами, яка б відстежувала залежності і автоматично встановлювала їх [14].

2.6 HTML і CSS

HTML (від англ. HyperText Markup Language - «мова гіпертекстової розмітки») - стандартна мова розмітки документів у Всесвітній павутині. Більшість вебсторінок містять опис розмітки на мові HTML (або XHTML). Мова HTML інтерпретується браузерами; отриманий в результаті інтерпретації форматований текст відображається на екрані монітора комп'ютера або мобільного пристрою [15].

Мова HTML є додатком SGML (стандартної узагальненої мови розмітки) і відповідає міжнародному стандарту ISO 8879.

Мова XHTML є більш суворим варіантом HTML, він слідує всім обмеженням XML і, фактично, XHTML можна сприймати як додаток мови XML до області розмітки гіпертексту.

У всесвітній павутині HTML-сторінки, як правило, передаються браузерам від сервера по протоколах HTTP або HTTPS, у вигляді простого тексту або з використанням шифрування [15].

Спочатку мова HTML була задумана і створена як засіб структурування та форматування документів без їх прив'язки до засобів відтворення (відображення). В ідеалі, текст з розміткою HTML повинен був без стилістичних та структурних спотворень відтворюватися на обладнанні з різною технічною оснащеністю (кольоровий екран сучасного комп'ютера, монохромний екран органайзера, обмежений за розмірами екран мобільного

телефону або пристрою і програми голосового відтворення текстів). Однак сучасне застосування HTML дуже далеко від його початкової задачі. Наприклад, тег <TABLE> призначений для створення в документах таблиць, але часто використовується і для оформлення розміщення елементів на сторінці. З плином часу основна ідея платформонезалежності мови HTML була принесена в жертву сучасним потребам в мультимедійному і графічному оформленні [15].

Текстові документи, що містять розмітку на мові HTML (такі документи зазвичай мають розширення .html або .htm), обробляються спеціальними додатками, які відображають документ в його форматованому вигляді. Такі додатки, звані «браузерами» або «інтернет-оглядачами», зазвичай надають користувачеві зручний інтерфейс для запиту вебсторінок, їх перегляду (і виведення на інші зовнішні пристрої) і, при необхідності, відправки введених користувачем даних на сервер. Найбільш популярними на сьогоднішній день браузерами є Google Chrome, Mozilla Firefox, Opera, Internet Explorer і Safari.

CSS (англ. Cascading Style Sheets - каскадні таблиці стилів) - формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки.

Переважно використовується як засіб опису, оформлення зовнішнього вигляду вебсторінок, написаних за допомогою мов розмітки HTML і XHTML, але може також застосовуватися до будь-яких XML-документах, наприклад, до SVG або XUL.

CSS використовується творцями вебсторінок для задання кольорів, шрифтів, розташування окремих блоків і інших аспектів представлення зовнішнього вигляду цих вебсторінок. Основною метою розробки CSS було розділення опису логічної структури вебсторінки (яке проводиться за допомогою HTML або інших мов розмітки) від опису зовнішнього вигляду цієї вебсторінки (яке тепер проводиться за допомогою формальної мови

CSS). Такий поділ може збільшити доступність документа, надати велику гнучкість і можливість управління його поданням, а також зменшити складність і повторюваність. Крім того, CSS дозволяє представляти один і той же документ в різних стилях або методах виведення, таких як екранне уявлення, друковане подання, читання голосом (спеціальним голосовим браузером або програмою читання з екрану), або при виведенні пристроями, що використовують шрифт Брайля [15].

Найбільш повно підтримують стандарт CSS є браузери, що працюють на двигунах Gecko (Mozilla Firefox і ін.), WebKit (Safari, Arora, Google Chrome) і Presto (Opera).

У минулому колись найпоширеніший браузер Internet Explorer 6 підтримує CSS далеко не повністю.

Навіть через 7 років після свого попередника Internet Explorer 7 хоча і значно поліпшив рівень підтримки CSS, але все ще містить значну кількість помилок.

В Internet Explorer 8 використовується новий двигун, який повністю підтримує CSS 2.1 і частково - CSS 3.

Для перевірки підтримки браузером вебстандартів (в тому числі і різних частин стандарту CSS) був розроблений тест Acid. Його друга версія називається Acid2, а третя, відповідно, Acid3.

Багато вебмайстрів для крос-браузерності стилів не використовують нововведення в CSS3, замінюючи їх зображеннями. Наприклад, замість заокруглення кутів використовують фонове зображення, на якому зображений цей блок без змісту (тексту) з заокругленими кутами.

Як і у випадку з JavaScript на сьогоднішній день немає альтернатив використанню HTML + CSS в області проектування і оформлення інтерфейсної частини вебдодатків [15]. Вибір мови програмування наведено у табл. 2.2.

Таблиця 2.2 – Обґрунтування вибору мови програмування

Критерій порівняння мов програмування	Мова програмування		
	CSP	JavaScript	HTML + CSS
Вебінтерфейс	+	+	+
Управління сесіями	+	–	–
Швидкість виконання коду	+	+–	+–
Багатопарадигмальність	+	+–	+
Кросплатформеність розробки	+	+–	+
Зручність використання	+	+–	+–

Отже, для реалізації програмного забезпечення здатного прогнозувати погоду для певного регіону, побудованого на базі машинного навчання нейронних мереж генетичними алгоритмами вибрано мову CSP.

3 РОЗРОБКА ОСНОВНИХ ПРОГРАМНИХ ЧАСТИН

Готова система являє собою загальнодоступний вебпродукт. Середовищем розробки була обрана платформа Cache Intersystems тому якості мови програмування для двигуна конструктора використовується мова CSP (внутрішня мова середовища розробки Cache Intersystems) для серверної складової, та JavaScript для частини, стосовної до розробки інтерфейсу. Дизайн вебсторінок був виконаний з використанням технологій HTML5 та CSS3 [12].

3.1 ER діаграма програми

ER діаграма – це діаграма даних, яка дозволяє описати концептуальну схему предметної області

Для кращого уявлення майбутньої програми, була створена ER-діаграма на рисунку 3.1.



Рисунок 3.1 – ER діаграма програми

На рисунку 3.1 зображено ER діаграму, на якій вказано усі залежності між усіма сутностями майбутньої програми.[6]

3.2 Алгоритм роботи програми

На рис 3.2 зображено алгоритм формування прогнозів погоди та етап оцінювання роботи алгоритмів прогнозування. А на рис. 3.3 зображено схему. Під час виконання програми ці два алгоритма виконуються паралельно та утворюють один повний цикл роботи програми.

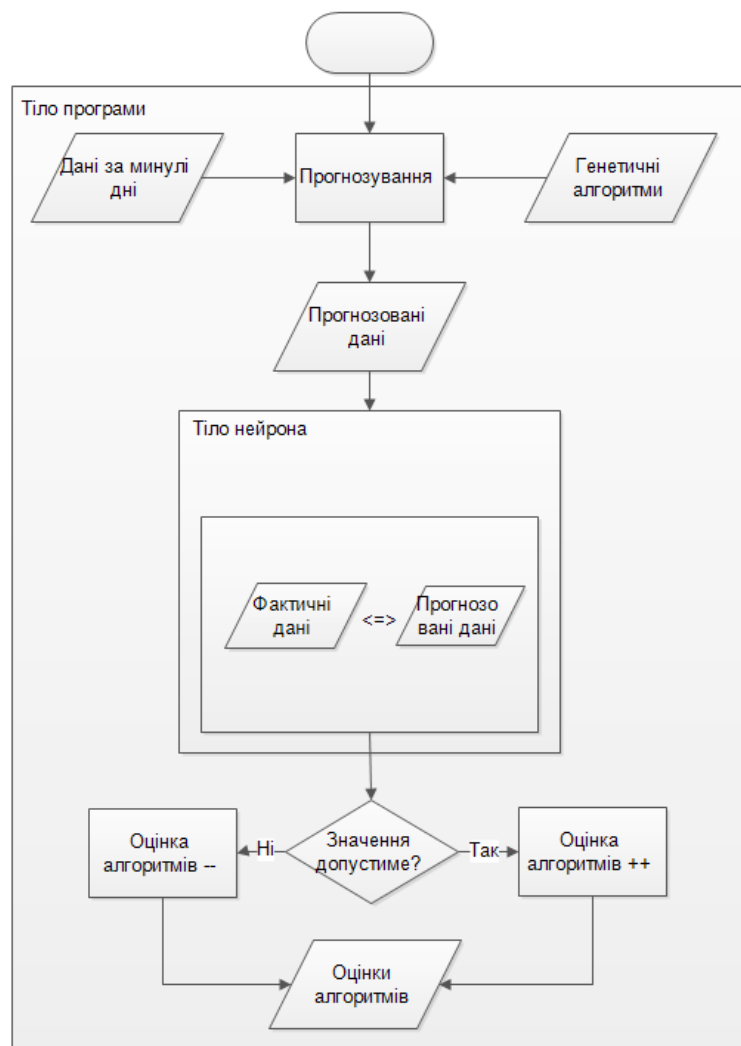


Рисунок 3.2 – Схема алгоритму формування прогнозів та оцінювання роботи алгоритмів прогнозування

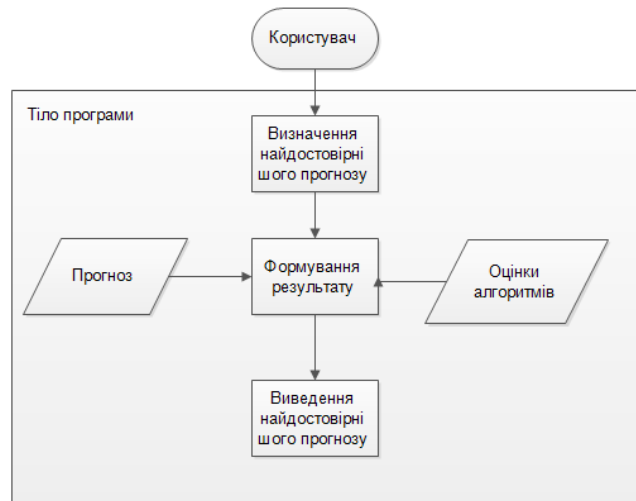


Рисунок 3.3 – Схема алгоритму формування результату на запит користувача

3.6 Структура середовища розробки

Cache Intersystems розробила не тільки ефективну БД з багаторівневим доступом до даних, але і адаптувала цілу платформу, з виділеним сервером, та власноруч розробленими мовами програмування для більш ефективного розроблення продуктів та зменшення часу відгуку між сервером та БД.

Caché надає програмістам свободу у виборі способу зберігання і доступу до даних через об'єкти, SQL або шляхом прямого доступу до багатовимірної структури даних. Незалежно від способу доступу, всі дані в базах даних Caché зберігаються в багатовимірних масивах.

Як тільки дані потрапляють в базу даних, всі способи доступу можуть бути використані одночасно для роботи над одними і тими ж даними.

Єдина архітектура даних (Unified Data Architecture) – унікальна особливість Caché. При описі класу бази даних Caché автоматично генерує реляційний опис тієї ж структури даних. Також, якщо було завантажено опис даних з реляційної бази даних (DDL) в словник даних, Caché автоматично створює реляційний і об'єктний опис завантажених структур даних. При

цьому всі визначення схем даних ведуться узгоджено, всі операції по редагуванню проводяться тільки з одним визначенням схеми даних.

Caché автоматично створює відображення для способу зберігання об'єктів і таблиць в багатовимірних структурах, або програміст може сам контролювати таке відображення.[1]

Переваги Caché:

- гнучкість: Всі способи доступу до одних і тих же даних (об'єктний, через SQL і багатовимірний) можуть використовуватися паралельно. Така гнучкість дає свободу вибору програмісту в визначенні способу доступу до даних, який найкращим чином підходить в його додатку;

- скорочення трудовитрат: Єдина архітектура даних Caché автоматично описує дані як класи і таблиці, використовуючи єдиний опис. При цьому не потрібна зміна коду, таким чином додатки розробляються і підтримуються з меншими затратами;

- використання традиційних навичок та існуючих додатків: Програмісти можуть використовувати досвід розробки додатків для реляційних СКБД і додавати новий об'єктний функціонал з розвитком існуючих додатків;

- підтримка NoSQL: Caché має властивості NoSQL завдяки наданню можливості прямого доступу до структур даних [2].

Висока ефективність бази даних Caché забезпечується на рівні ядра підтримкою багатовимірних структур даних. Вбудовані скриптові мови працюють безпосередньо з багатовимірними структурами, забезпечуючи додаткам найвищу продуктивність і найширший діапазон можливостей в організації сховища даних. Багато додатків реалізовані повністю із застосуванням такого прямого доступу до даних. Підхід з використанням прямого «глобального доступу» прийнято застосовувати там, де з метою досягнення найвищої продуктивності проектуються нетрадиційні або

спеціалізовані структури даних і не потрібно забезпечувати об'єктний або реляційний інтерфейс до цих даних.

Для багатовимірного ядра не існує словників даних і, відповідно, немає описів структур даних.

В нашій БД був застосований багаторівневий доступ до даних за допомогою NoSQL технології, таким чином було створено 3 так званих глобала [7].

Глобал - це спеціальний спосіб зберігання і обробки даних, зовсім інший, ніж таблиці в SQL. Вони з'явилися в 1966 році в мові М (UMPS) (еволюційний розвиток - Caché ObjectScript, далі COS) в медичних БД і до сих пір там активно використовуються, а також проникли до деяких інших областей, де потрібна надійність і висока продуктивність: фінанси, трейдинг і т.д. [8].

Тож для надійної роботи нашої інформаційно-довідкової системи було створено 4 глобала, із назвами та призначенням глобалів можна ознайомитися в таблиці 3.1.

Таблиця 3.1 – Назви та призначення глобалів

Назва глобала	Призначення глобала
Prognoz	Список прогнозів
Rate	Список оцінок алгоритмів
Weathertab	Список фактичних даних для аналізу
Fact	Список фактичних даних для порівняння з прогнозом

Ознайомившись із призначенням кожного глобала, можна розглянути їх структуру. Структура глобалів Prognoz, Rate, наведена у таблицях 3.2, 3.3 відповідно.

Таблиця 3.2 – структура глобала Prognoz

Поле	Тип	Призначення
id	int	Інкрементальний ключовий ідентифікатор
day	varchar	День на який зроблено прогноз
time	vacrchar	Час на який зроблено прогноз
type	varchar	Тип прогнозованого значення
value	varchar	Прогнозоване значення
otbor	varcha	Алгоритм відбору, що був застосований
chrest	varcha	Алгоритм схрещування, що був застосований
mutation	varcha	Алгоритм мутації, що була застосований

Таблиця 3.3 – структура глобала Rate

Поле	Тип	Призначення
Chrest_arifm	int	Поле оцінки алгоритму арифметичного схрещування
Chrest_odn	Int	Поле оцінки алгоритму одиничного схрещування
Chrest_ravnom	Int	Поле оцінки алгоритму рівномірного схрещування
Chrest_sravn	Int	Поле оцінки алгоритму порівняльного схрещування
Mut_gauss	Int	Поле оцінки алгоритму нормальної мутації Гаусса
Mut_prost	Int	Поле оцінки алгоритму простої мутації
Mut_sluch	Int	Поле оцінки алгоритму випадкової мутації
Mut_tochk	Int	Поле оцінки алгоритму точкової мутації
Por_otbor	Int	Поле оцінки алгоритму порогового відбору
Prop_otbor	Int	Поле оцінки алгоритму пропорційного відбору
Range_otbor	Int	Поле оцінки алгоритму відбору ранжируванням
Turn_otbor	Int	Поле оцінки алгоритму турнірного відбору

Структура глобалів Weathertab та Fact наведена у таблицях 3.4, 3.5 відповідно.

Таблиця 3.4 – структура глобала Weathertab

Поле	Тип	Призначення
Id	Int	Інкрементальний ключовий ідентифікатор
day	varchar	День із фактичними даними
time	Varchar	Час із фактичними даними
temp	Varchar	Фактична температура
Po	Varchar	Фактичний тиск на рівні метеостанції
P	Varchar	Фактичний середній тиск
U	Varchar	Фактична відносна вологість
DD	Varchar	Фактичний напрямок вітру
Ff	Varchar	Фактична швидкість вітру
N	Varchar	Фактична хмарність
WW	varchar	Фактична погода, яку віщає метеостанція

Таблиця 3.5 – структура глобала Fact

Поле	Тип	Призначення
day	varchar	Дата прогнозованого дня
time	Varchar	Час прогнозованого дня
temp	Varchar	Температура прогнозованого дня
Po	Varchar	тиск на рівні метеостанції прогнозованого дня
P	Varchar	середній тиск прогнозованого дня
U	Varchar	відносна вологість прогнозованого дня
DD	Varchar	напрямок вітру прогнозованого дня
Ff	Varchar	швидкість вітру прогнозованого дня
N	Varchar	Хмарність прогнозованого дня
WW	varchar	погода, яку віщає метеостанція прогнозованого дня

Алгоритми прогнозування в програмі були представлені у вигляді функцій. Таким чином робота алгоритму порогового відбору була описана функцією `por_otbor()` на мові Javascript, описання усіх розроблених функцій та їх призначення можна побачити в таблиці 3.6.

Таблиця 3.6 - Описання та призначення розроблених функцій

Функція	Призначення
1	2
<code>Chrest_arifm()</code>	Функція, що описує роботу алгоритму арифметичного схрещування
<code>Chrest_odn()</code>	Функція, що описує роботу алгоритму одиничного схрещування
<code>Chrest_ravnom()</code>	Функція, що описує роботу алгоритму рівномірного схрещування
<code>Chrest_sravn()</code>	Функція, що описує роботу алгоритму порівняльного схрещування
<code>Mut_gauss()</code>	Функція, що описує роботу алгоритму нормальної мутації Гаусса
<code>Mut_prost()</code>	Функція, що описує роботу алгоритму простої мутації
<code>Mut_sluch()</code>	Функція, що описує роботу алгоритму випадкової мутації
<code>Mut_tochk()</code>	Функція, що описує роботу алгоритму точкової мутації
<code>Por_otbor()</code>	Функція, що описує роботу алгоритму порогового відбору
<code>Prop_otbor()</code>	Функція, що описує роботу алгоритму пропорційного відбору
<code>Range_otbor()</code>	Функція, що описує роботу алгоритму відбору ранжируванням
<code>Turn_otbor()</code>	Функція, що описує роботу алгоритму турнірного відбору
<code>Prognoz()</code>	Функція, реалізація якої прогнозує дані та вносить їх у БД

Продовження таблиці 3.6

1	2
Rounds()	Функція, призначенням якої є просіювання даних через фіксовану кількість циклів
Parts()	Функція, яка описує реалізації скомпонованих між собою алгоритмів відбору, схрещування та мутації

В цьому розділі було детально розглянуто діаграму сутність-зв'язок, алгоритми роботи програми, та мови програмування, застосування яких допоможе зробити інформаційну систему функціональною, швидкодіюною, а інтерфейс системи простим та зрозумілим.

4 ТЕСТУВАННЯ ТА ДОСЛІДЖЕННЯ ВИГОТОВЛЕННОЇ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Інформаційний застосунок «Прогноз погоди» призначена для користувачів, які бажають дізнатися прогноз погоди певної місцевості на наступний день.

Інформаційний застосунок «Прогноз погоди» несе виключно інформаційний характер із мінімальним втручанням користувача у роботу програми, так як головною задачею системи є надання максимально достовірної інформації вплив на яку може мати лише програміст.

4.2 Характеристики програми

Програма виконана за допомогою мови програмування Cache Server Page та інтегрованої середовища програмування Cache Studio. Після оновлення функціоналу треба перезапустити сервер.

До складу технічних засобів повинен входити IBM-сумісна персональна електронна обчислювальна машина (ПЕОМ), що виконує роль сервера, що включає в себе:

- процесор Pentium-2.0Hz, не менше;
- оперативну пам'ять об'ємом, 1Гігабайт, не менше;
- HDD, 20 Гігабайт, не менше;
- операційну систему Microsoft Windows XP/Vista/7 або більш пізні версії;
- TryCache server (Cache intersystem Server).

Налаштування сервера відбувається за допомогою спеціального інсталятора, який можна знайти на офіційному сайті Intersystems Cache. Після того, як інсталятор був скачаний, його необхідно запустити та у

декілька простих дій встановити його на машину, яка буде виступати у ролі сервера:

- Дія 1: Погодитися із ліцензійними вимогами програми обрати перемикач «Я приймаю...» та натиснути кнопку «Далі» рис 4.1;
- Дія 2: Обрати директорію для встановлення пакету програм та натиснути кнопку «Далі» рис 4.1;
- Дія 3: Натиснути кнопку «Встановити», тим самим підтвердити початок інсталяції програми на комп'ютер рис 4.2;
- Дія 4: Після того як процес установки пакету програм закінчиться-зняти галочку біля тексту «Показати документ...» та натиснути кнопку «Готово» рис 4.2.

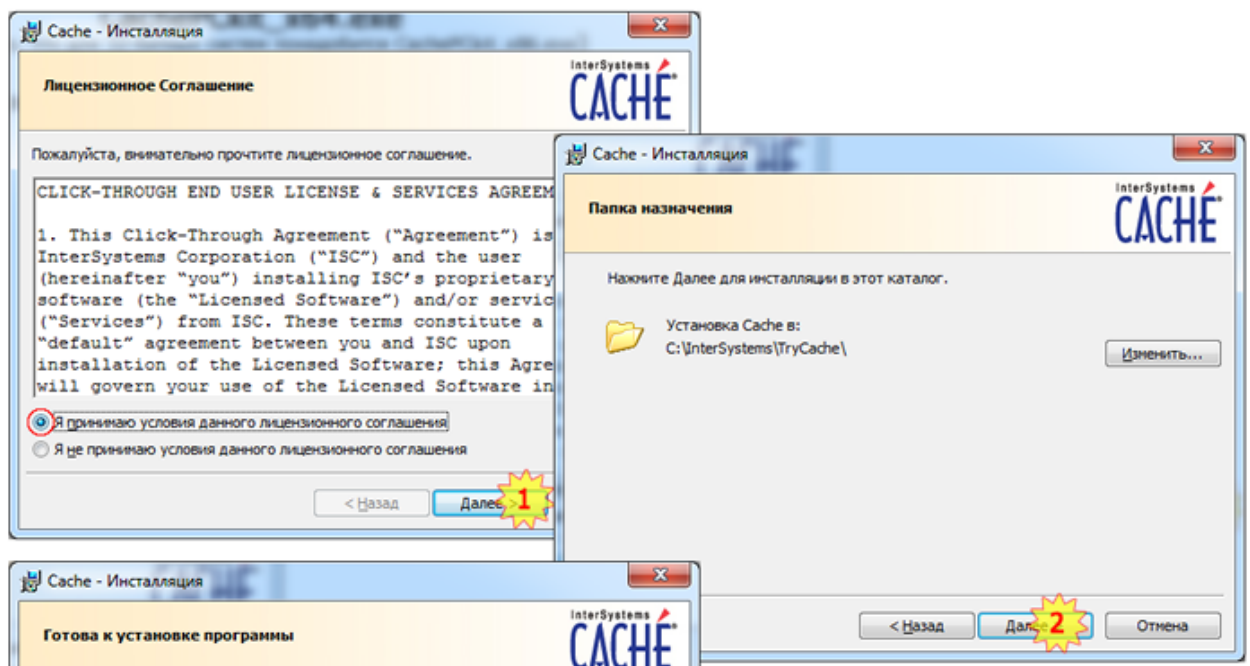


Рисунок 4.1 – Дія 1 і Дія 2 установки пакету програм Cache Intersystems

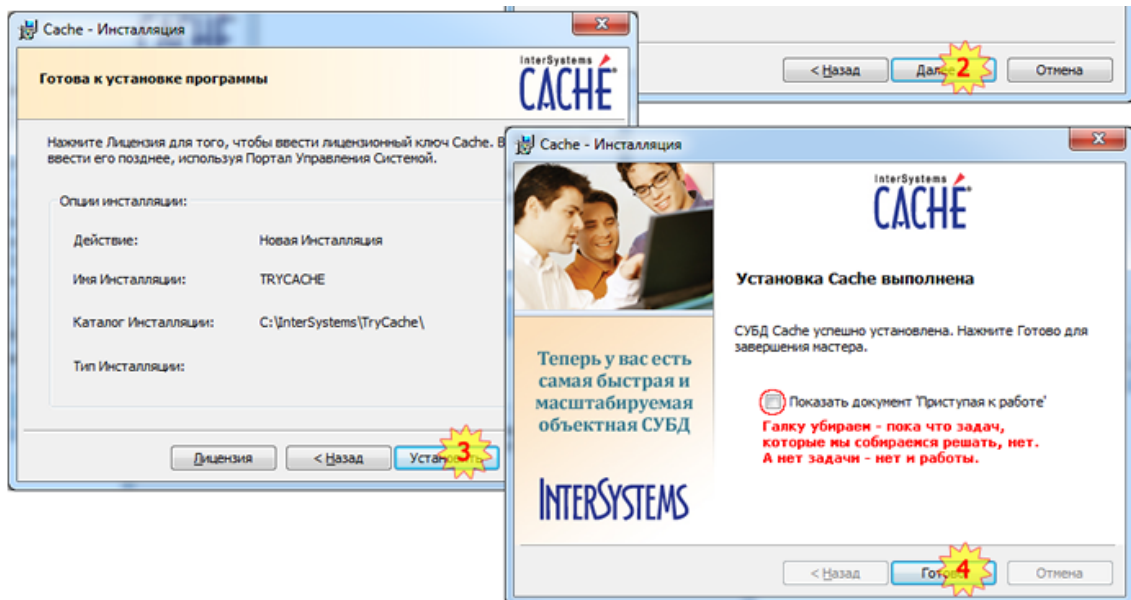


Рисунок 4.2 – Дія 3 і Дія 4 установки пакету програм Cache InterSystems

4.3 Звертання до програми

Звертання до програми відбувається через вікно TRYCache Studio рис.4.3. Звідти можна редагувати увесь код, який стосується програми, в тому числі і функції, які були описані у розділі 3.

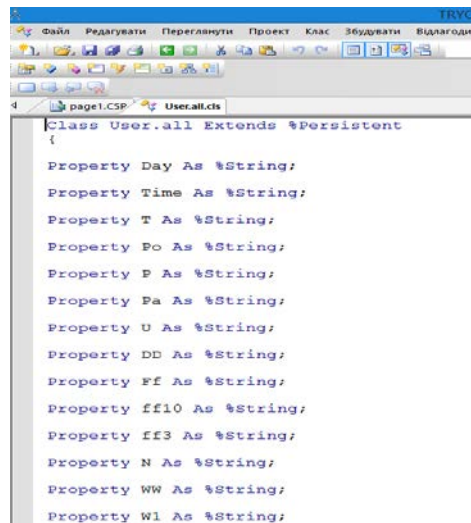


Рисунок 4.3 – Вікно TRYCache Studio

Звертання до БД відбувається через термінал Cache рис.4.4 або через TRYCache Studio рис. 4.3.

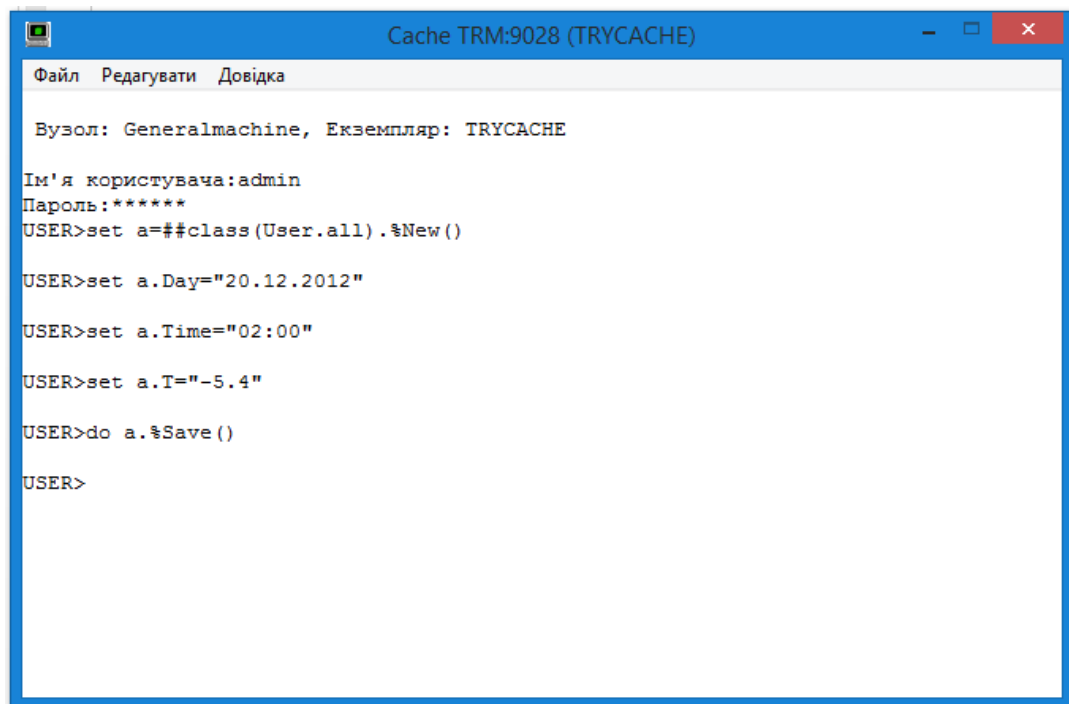
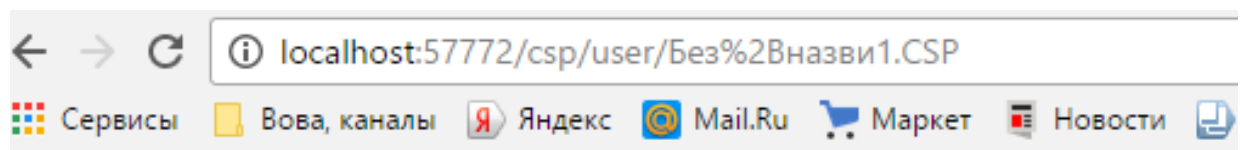


Рис 4.4 – Термінал Cache

4.4 Повідомлення

Якщо під час компілювання були вказані невірні дані форми, та вони не могли бути розпізнані виведеться вікно помилки рис. 4.5, якщо була допущена помилка у назвах функцій чи частинах інтерпретатора коду буде виведено вікно помилки 4.6.



Not Found

The requested URL /csp/user/Без+назви1.CSP was not found on this server.

Рисунок 4.5 – Вікно помилки розпізнавання

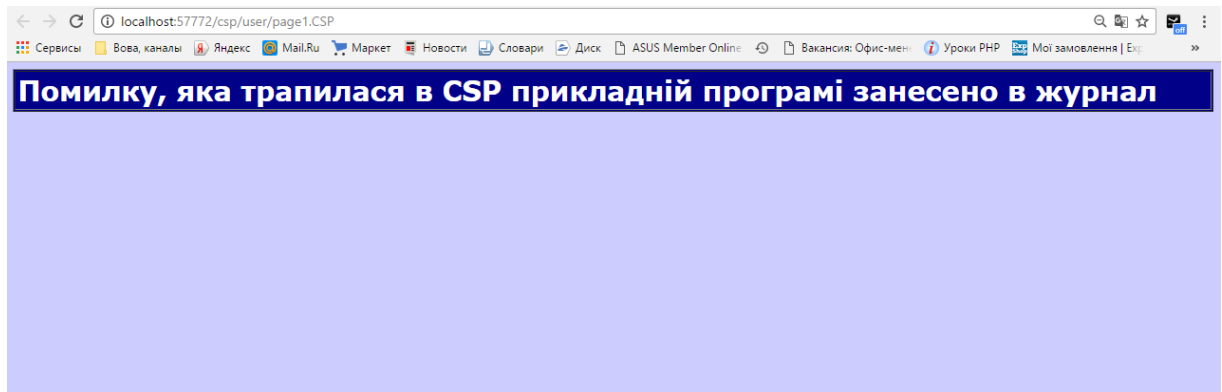


Рисунок 4.6 – Вікно помилки інтерпретатору коду

4.5 Виконання програми

Для демонстрації результатів роботи нижче наведені скріншоти з основними сценаріями. Спочатку користувач вибирає місто, в якому його цікавить погода рис. 4.7.

Так як наш продукт класифікується як інформаційний застосунок, то втручання користувача в роботи програми зведено до мінімуму, що має на увазі: коли користувач заходить за адресою вебпродукта він потрапляє на головну сторінку рис. 4.8, де для нього відображені вже спрогнозовані дані на наступний день у вигляді таблиці.

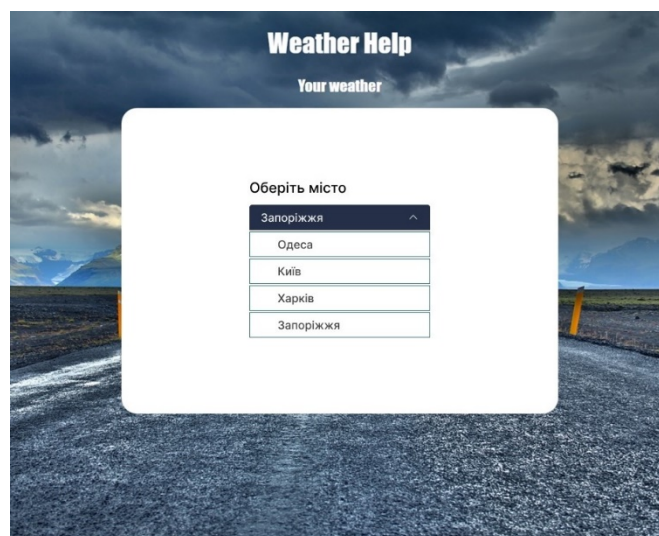


Рисунок 4.7 – Сторінка вибору міста



Рисунок 4.8 – Головна сторінка програмного продукту

Також додався запис про користувача до таблиці рис.4.9.

+ Параметри

	id	firstname	lastname	city	date	changeFlag
☐	354303518	Михаил	Щербак	Запорожье	2020-05-05	NULL

↑ ☐ Отметить все С отмеченными:

Рисунок 4.9 – – Запис про користувача

4.6 Порівняння інтелектуальних методів для прогнозування погоди

Коли модель пройшла 3 цикли, були отримані оцінки роботи алгоритмів, які були задіяні у прогнозуванні даних.

На рис. 4.10 зображено гістограму, яка показує до скількох достовірних прогнозів причетний кожен алгоритм відбору.

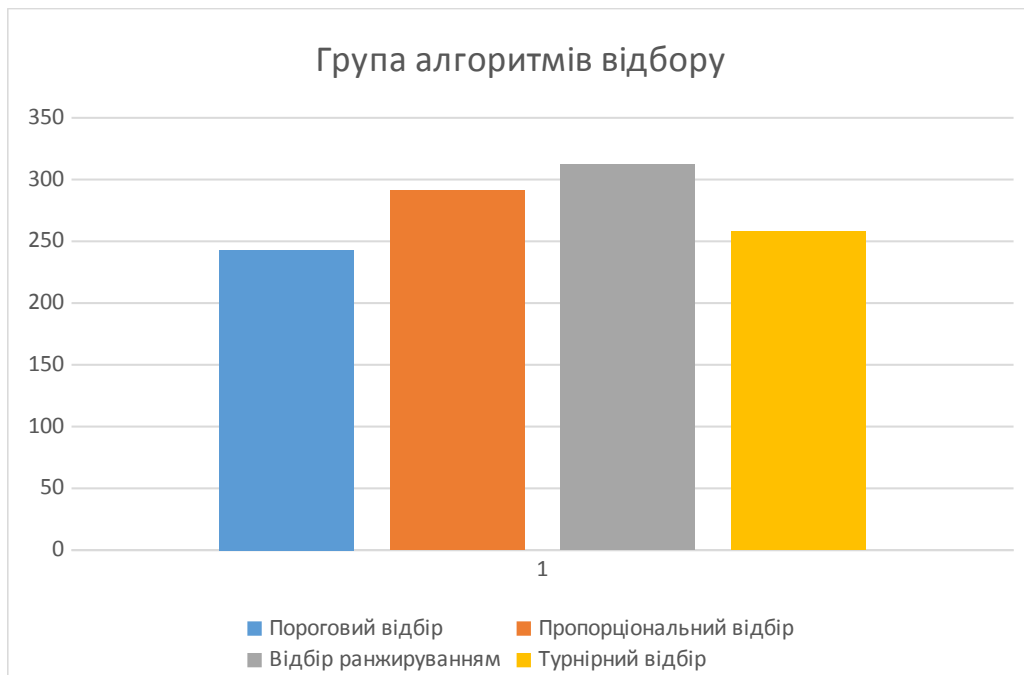


Рисунок 4.10 – Гістограма алгоритмів відбору

На рис 4.11 зображено гістограму, яка показує до скількох достовірних прогнозів причетний кожен алгоритм схрещування.

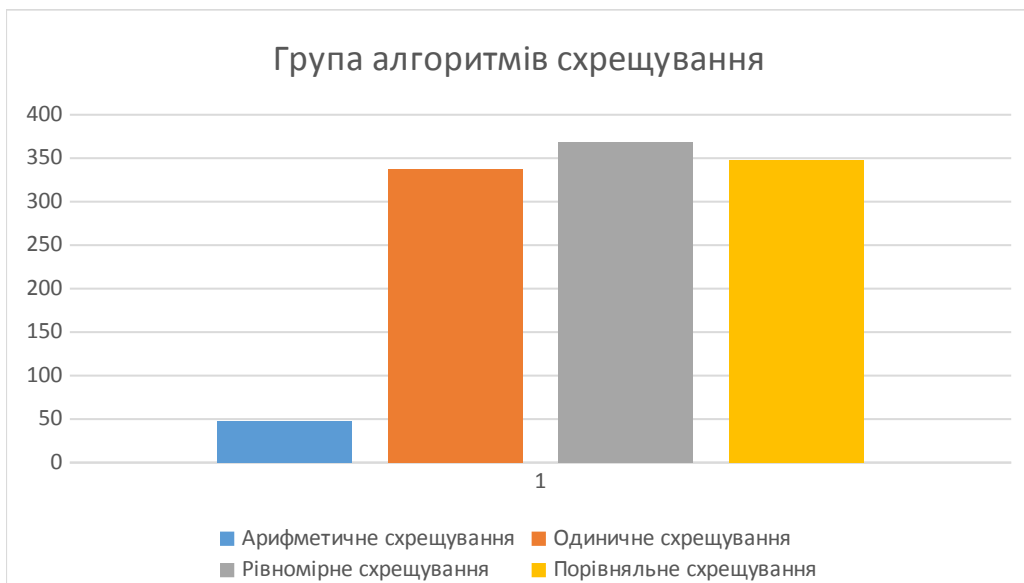


Рисунок 4.11 – Гістограма алгоритмів схрещування

І нарешті на рис 4.12 зображено гістограму, яка показує до скількох достовірних прогнозів причетний кожен алгоритм мутації.

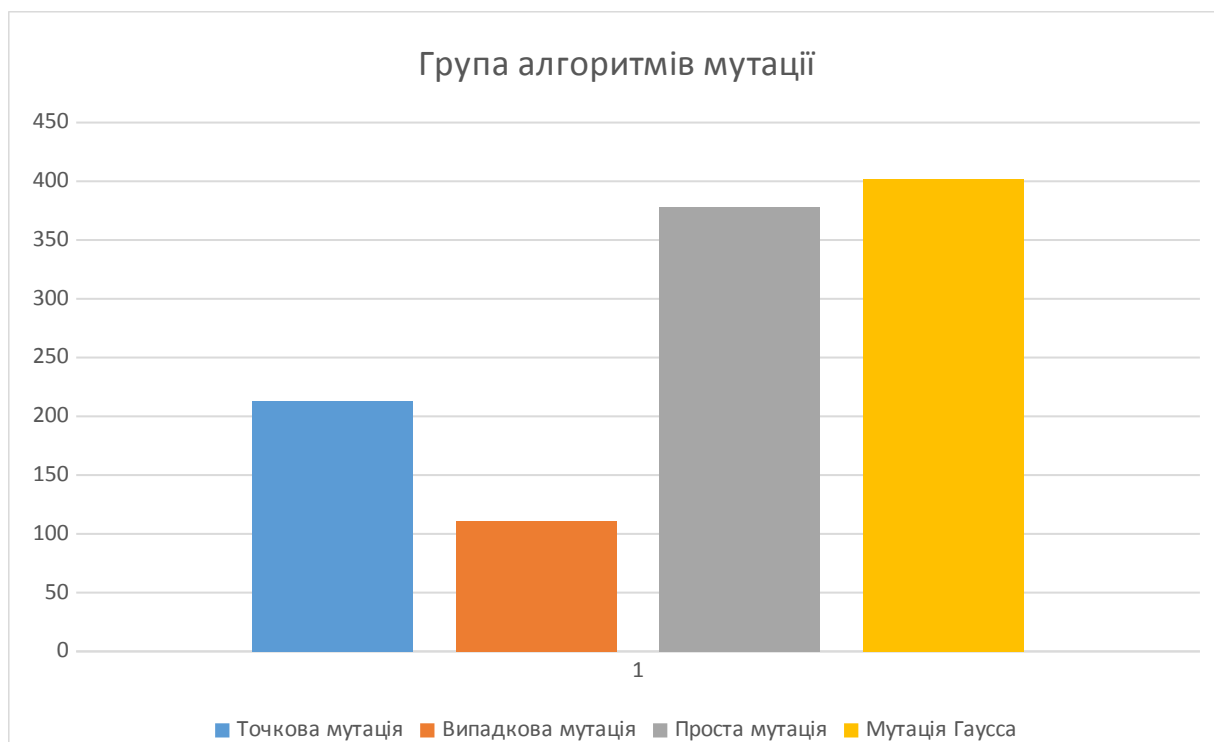


Рисунок 4.12 – Гістограма алгоритмів мутації

В програмі реалізований новітній спосіб прогнозування, в той час як сама програма надає інформацію користувачу про найдостовірніші спрогнозовані дані, у нашому випадку це прогноз погоди для певної місцевості, що буде на наступний день. Інтерфейс програми простий і зрозумілий. Інформація отримана з рис. 4.12, 4.11 та рис 4.12 показує що результати роботи алгоритмів відрізняються між собою, тому можна зробити висновок, що програма працює справно.

ВИСНОВКИ

Розроблена програма відповідає усім основним вимогам для ефективної роботи інформаційного застосунку «Прогноз погоди». Реалізована нейронна мережа, навчена генетичними алгоритмами, являє собою приклад елементарного штучного інтелекту, здатного до самонавчання. Пропрацювавши деякий час, система видала результати ефективності кожного алгоритма з групи алгоритмів відбору, алгоритмів схрещування, та алгоритмів мутації, що дозволило визначити який був найефективнішим у своїй групі.

Отримані результати демонструють нам ефективність обраної нами нейронної мережі, тому кожного разу перед тим як продемонструвати користувачу прогноз на новий день, програма буде обирати прогноз, у якому були задіяні найефективніші представники своєї групи.

Були досягнені наступні цілі:

- проаналізовано аналогічні системи засновані на машинному навчанні нейронних мереж;
- вивчено архітектуру та можливості середовища TruCache;
- розроблено методи, які засновані на можливостях і особливостях генетичних алгоритмів та інтерпретовано їх під машинний код;
- розроблено методи, які засновані на можливостях і особливостях нейронних мереж та інтерпретовано їх під машинний код;
- навчено нейронні мережі генетичними алгоритмами;
- продемонстровано можливості розробленого продукту на прикладі прогнозування погоди.

Проведені експерименти засвідчили, що програма має достатню працездатність і може використовуватися за призначенням для прогнозування погоди.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Приклади інформаційних систем [Електрон. ресурс]. – Режим доступу: https://miyklas.com.ua/p/informatica/5-klas/informatciini-protcesi-ta-sistemi-39831/informatciini-protcesi-ta-sistemi-rol-informatciinikh-tekhnologii-u-zhitt_-39902/re-c048319d-e1b4-44f9-92c6-fed06733d8b5/.
2. Комп'ютерні системи [Електрон. ресурс]. – Режим доступу: https://nubip.edu.ua/sites/default/files/u34/konspekt_ks.pdf/.
3. Структура інформаційної системи [Електрон. ресурс]. – Режим доступу: http://www.ck.vk.mnau.edu.ua/ck/portfolio_Macovey/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_IKT/%D0%A1%D0%B0%D0%BC_%D1%80%D0%BE%D0%B1/%D1%81%D0%B0%D0%BC_%D1%80%D0%BE%D0%B11.htm.
5. Meteo [Електрон. ресурс]. – Режим доступу: <https://meteo.ua/ua>.
6. Meteoprog [Електрон. ресурс]. – Режим доступу: <https://www.meteoprog.com/ua/weather/Zaporizhzhia/>.
7. Gismeteo [Електрон. ресурс]. – Режим доступу: <https://www.gismeteo.ua/>.
8. Чисельне метеорологічне прогнозування [Електрон. ресурс]. – Режим доступу: <http://cloud-5.bitp.kiev.ua/?p=1643>.
9. Чисельне метеорологічне прогнозування [Електрон. ресурс]. – Режим доступу: <http://cloud-5.bitp.kiev.ua/?p=1643>.
10. Генетичні алгоритми. Ключові поняття і методи реалізації [Електрон. ресурс]. – Режим доступу: http://www.znannya.org/?view=ga_general.
11. Руководство по MySQL [Електрон. ресурс]. – Режим доступу: <https://metanit.com/sql/mysql/>.

12. Connect to an Oracle database with Power BI Desktop [Electronic resource]. – Access mode: <https://learn.microsoft.com/en-us/power-bi/connect-data/desktop-connect-oracle-database>.

13. Using Caché Server Pages (CSP) [Electronic resource]. – Access mode: <https://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GCSP>.

14. JavaScript [Electronic resource]. – Access mode: <https://developer.mozilla.org/>.

15. Електронний HTML і CSS довідник [Електрон. ресурс]. – Режим доступу: <https://html-css.co.ua/>.

ДОДАТОК А

Програмний код компонентів сайту

```

<html>
<head>

<!-- Put your page Title here -->
<title>         Cache Server Page </title>
<script type="text/javascript"
src="http://scriptjava.net/source/scriptjava/scriptjava.js"></script>

</head>

<BODY bgcolor="#FFFFFF" onload="razb();send_family();">

<script language="Cache" runat="server">

set v=$get(%request.Data("Date",1))
if (v != ""){
w "<!--"
}
</script>

<script language="JavaScript" type="text/javascript">

        pat();

</script>

<script language="Cache" runat="server">

set v=$get(%request.Data("Date",1))
if (v != ""){
w "-->"
}

</script>

<!-- Define a search function -->

<form id="mnform" name="theForm">
<table cellpadding="5">
<tr>
<td align="right"><b>Вкажіть день:</b></td>
<td><input id = "Data" name="Data" size="8" type="date" ></td>
<td align="right"></td>
<td><input id="subm" name="Go" type="submit" value="Go" onclick="razb();"></td>
</tr>
</table>
</form>

<csp:object OBJID='#($get(%request.Data("Day",1)))#' classname="User.all" name="day">

<form id="pstform" name="Form2" >

```

```

    <input id = "Date" name="Day" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("Day",1))))#>
    <input id = "d_o_d" name="day_of_date" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("Date",1))))#>
    <input id = "m_o_d" name="month_of_date" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("Date",2))))#>
    <input id = "y_o_d" name="year_of_date" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("Date",3))))#>

```

```

<input id="GET" name="GET" type="submit" value="GET" >
</form>

```

```

<!--<form id="year_form" name="Form_year" >
    <input id = "year05" name="year05" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("year05",1))))#>
    <input id = "year06" name="year06" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("year06",1))))#>
    <input id = "year07" name="year07" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("year07",1))))#>
    <input id = "year08" name="year08" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("year08",1))))#>
    <input id = "year09" name="year09" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("year09",1))))#>
    <input id = "year10" name="year10" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("year10",1))))#>
    <input id = "year11" name="year11" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("year11",1))))#>
    <input id = "year12" name="year12" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("year12",1))))#>
    <input id = "year13" name="year13" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("year13",1))))#>
    <input id = "year14" name="year14" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("year14",1))))#>
    <input id = "year15" name="year15" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("year15",1))))#>
    <input id = "year16" name="year16" size="8" type="input"
value='#(..EscapeHTML($get(%request.Data("year16",1))))#>
</form>-->

```

```

<script language="JavaScript" type="text/javascript">

```

```

/*function pat(){
    //document.getElementById("subm").click();
    var url = document.URL;
    var subs = "day_of_date";
    if (url.includes(subs)){
        var vars = url.split('&');
        alert(vars[2]);
    }
}

```

```

function swapEl(event,elem){
    elem.parentNode.insertBefore(elem,elem.parentNode.firstChild);
}

/*function asda(){
    document.getElementById("GET").click();
}*/

function razb(){
    var day =
'#(..EscapeHTML($get(%request.Data("Data",1))))#';//document.getElementById("Date").value;

    //var date = document.getElementById("Data").value;

    if (day != ""){
        var spl = day.split('-');
        //alert(day);
        //alert(spl[0] + " " + spl[1] + " " + spl[2]);
        //alert( "Привет" );
        //alert(day);
        document.getElementById("Date").value = spl[2] + "." + spl[1] + "." + spl[0];
        document.getElementById('d_o_d').value = spl[2];
        document.getElementById('m_o_d').value = spl[1];
        document.getElementById('y_o_d').value = spl[0];
        /*var y05=spl[0]-11;
        document.getElementById('year05').value = spl[2] + "." + spl[1] + "." + y05;
        var y06=spl[0]-10;
        document.getElementById('year06').value = spl[2] + "." + spl[1] + "." + y06;
        var y07=spl[0]-9;
        document.getElementById('year07').value = spl[2] + "." + spl[1] + "." + y07;
        var y08=spl[0]-8;
        document.getElementById('year08').value = spl[2] + "." + spl[1] + "." + y08;
        var y09=spl[0]-7;
        document.getElementById('year09').value = spl[2] + "." + spl[1] + "." + y09;
        var y10=spl[0]-6;
        document.getElementById('year10').value = spl[2] + "." + spl[1] + "." + y10;
        var y11=spl[0]-5;
        document.getElementById('year11').value = spl[2] + "." + spl[1] + "." + y11;
        var y12=spl[0]-4;
        document.getElementById('year12').value = spl[2] + "." + spl[1] + "." + y12;
        var y13=spl[0]-3;
        document.getElementById('year13').value = spl[2] + "." + spl[1] + "." + y13;
        var y14=spl[0]-2;
        document.getElementById('year14').value = spl[2] + "." + spl[1] + "." + y14;
        var y15=spl[0]-1;
        document.getElementById('year15').value = spl[2] + "." + spl[1] + "." + y15;
        var y16=spl[0];
        document.getElementById('year16').value = spl[2] + "." + spl[1] + "." + y16;*/
        document.getElementById("GET").click();
    }
}

```

</script>

```
<input type="submit" name="update" onclick="d();" value="update" >
```

```
<script language="Cache" runat="server">
w "<p>мммвапвап</p>"
```

```
</script>
<script language="JavaScript" type="text/javascript">
function d() {
location.reload(true)
}
</script>
```

```
<script language="Cache" runat="server">
set dDay="Day"
set type="text"
set value="blabla"
//w "<input name="",dDay,"" type="",type,"" value="",value,"">"
/*set value=$get(%request.Data("Day",1))
set year05=$get(%request.Data("year05",1))
set year06=$get(%request.Data("year06",1))
set year07=$get(%request.Data("year07",1))
set year08=$get(%request.Data("year08",1))
set year09=$get(%request.Data("year09",1))
set year10=$get(%request.Data("year10",1))
set year11=$get(%request.Data("year11",1))
set year12=$get(%request.Data("year12",1))
set year13=$get(%request.Data("year13",1))
set year14=$get(%request.Data("year14",1))
set year15=$get(%request.Data("year15",1))
set year16=$get(%request.Data("year16",1))*/
//set val1=$PIECE(value,"-",1)
//set val2=$PIECE(value,"-",2)
//set val3=$PIECE(value,"-",3)
//set dat=val1+"."+val2+"."+val3
//w "<p>"+val1+"."+val2+"."+val3+"</p>"
//set val = $TRANSLATE(value,"-",".")
/*set suc="0"
set suc05="0"
set suc06="0"
set suc07="0"
set suc08="0"
set suc09="0"
set suc10="0"
set suc11="0"
set suc12="0"
set suc13="0"
set suc14="0"
set suc15="0"
set suc16="0"*/
set k = 1
for i=1:1:204{
//while (k<17 && i=i1){
```



```

        if a.Time = "17:00" {
            set %request.Data(k,1) = a.T
            set k = k+1
        }
        //w "<p>",i," ",%request.Data("val"+i,k),"</p>"
        //w "<p>val",i," ", %request.Data("val"+i,k),"</p>"
    }
    set k = 121
    for i=1:1:204{
        //while (k<17 && i'=i1){
        set a=##class(User.all).%OpenId(i)
            if a.Time = "20:00" {
                set %request.Data(k,1) = a.T
                set k = k+1
            }
            //w "<p>",i," ",%request.Data("val"+i,k),"</p>"
            //w "<p>val",i," ", %request.Data("val"+i,k),"</p>"
        }
        set k = 141
        for i=1:1:204{
            //while (k<17 && i'=i1){
            set a=##class(User.all).%OpenId(i)
                if a.Time = "23:00" {
                    set %request.Data(k,1) = a.T
                    set k = k+1
                }
                //w "<p>",i," ",%request.Data("val"+i,k),"</p>"
                //w "<p>val",i," ", %request.Data("val"+i,k),"</p>"
            }
        }

        /*set k = 1
        for i=95:8:221{
            //while (k<17 && i'=i1){
            set a=##class(User.all).%OpenId(i)
            w "<p>day ",i," ",a.Day,"</p>"
            w "<p>day ",i," ",a.Time,"</p>"
            set %request.Data("val"+i,k) = a.T
            w %request.Data("val"+i,k)
            w "val",i," ", %request.Data("val"+i,k)
        }
        set k = 1
        for i=96:8:221{
            //while (k<17 && i'=i1){
            set a=##class(User.all).%OpenId(i)
            w "<p>day ",i," ",a.Day,"</p>"
            w "<p>day ",i," ",a.Time,"</p>"
            set %request.Data("val"+i,k) = a.T
            w %request.Data("val"+i,k)
            w "val",i," ", %request.Data("val"+i,k)
        }
        set k = 1
        for i=97:8:221{
            //while (k<17 && i'=i1){
            set a=##class(User.all).%OpenId(i)
            w "<p>day ",i," ",a.Day,"</p>"
            w "<p>day ",i," ",a.Time,"</p>"

```

```

        set %request.Data("val"+i,k) = a.T
        w %request.Data("val"+i,k)
        w "val",i," ", %request.Data("val"+i,k)
    }
    set k = 1
    for i=98:8:221{
        //while (k<17 && i'=i1){
        set a=##class(User.all).%OpenId(i)
        w "<p>day ",i," ",a.Day,"</p>"
        w "<p>day ",i," ",a.Time,"</p>"
        set %request.Data("val"+i,k) = a.T
        w %request.Data("val"+i,k)
        w "val",i," ", %request.Data("val"+i,k)
    }
    set k = 1
    for i=99:8:221{
        //while (k<17 && i'=i1){
        set a=##class(User.all).%OpenId(i)
        w "<p>day ",i," ",a.Day,"</p>"
        w "<p>day ",i," ",a.Time,"</p>"
        set %request.Data("val"+i,k) = a.T
        w %request.Data("val"+i,k)
        w "val",i," ", %request.Data("val"+i,k)
    }
    set k = 1
    for i=100:8:221{
        //while (k<17 && i'=i1){
        set a=##class(User.all).%OpenId(i)
        w "<p>day ",i," ",a.Day,"</p>"
        w "<p>day ",i," ",a.Time,"</p>"
        set %request.Data("val"+i,k) = a.T
        w %request.Data("val"+i,k)
        w "val",i," ", %request.Data("val"+i,k)
    }
    set k = 1
    for i=101:8:221{
        //while (k<17 && i'=i1){
        set a=##class(User.all).%OpenId(i)
        w "<p>day ",i," ",a.Day,"</p>"
        w "<p>day ",i," ",a.Time,"</p>"
        set %request.Data("val"+i,k) = a.T
        w %request.Data("val"+i,k)
        w "val",i," ", %request.Data("val"+i,k)
    }
    set k = 1
    for i=102:8:221{
        //while (k<17 && i'=i1){
        set a=##class(User.all).%OpenId(i)
        w "<p>day ",i," ",a.Day,"</p>"
        w "<p>day ",i," ",a.Time,"</p>"
        set %request.Data("val"+i,k) = a.T
        w %request.Data("val"+i,k)
        w "val",i," ", %request.Data("val"+i,k)
    }
    set k = 1
    for i=103:8:221{

```

```

//while (k<17 && i'=i1){
set a=##class(User.all).%OpenId(i)
w "<p>day ",i," ",a.Day,"</p>"
w "<p>day ",i," ",a.Time,"</p>"
set %request.Data("val"+i,k) = a.T
w "val",i," ", %request.Data("val"+i,k)
}
*/
</script>

```

```

<p><input id="send_res" value="send" type="button" method="post"></p>
</form>

```

```

<script language="JavaScript" type="text/javascript">
var val1_2 = (Number(document.getElementById('val1_2').value) * 1.8 + 32).toFixed(0);
var val2_2 = (Number(document.getElementById('val2_2').value) * 1.8 + 32).toFixed(0);
var val3_2 = (Number(document.getElementById('val3_2').value) * 1.8 + 32).toFixed(0);
var val4_2 = (Number(document.getElementById('val4_2').value) * 1.8 + 32).toFixed(0);
var val5_2 = (Number(document.getElementById('val5_2').value) * 1.8 + 32).toFixed(0);
var val6_2 = (Number(document.getElementById('val6_2').value) * 1.8 + 32).toFixed(0);
var val7_2 = (Number(document.getElementById('val7_2').value) * 1.8 + 32).toFixed(0);
var val8_2 = (Number(document.getElementById('val8_2').value) * 1.8 + 32).toFixed(0);
var val9_2 = (Number(document.getElementById('val9_2').value) * 1.8 + 32).toFixed(0);
var val10_2 = (Number(document.getElementById('val10_2').value) * 1.8 + 32).toFixed(0);
var val11_2 = (Number(document.getElementById('val11_2').value) * 1.8 + 32).toFixed(0);
var val12_2 = (Number(document.getElementById('val12_2').value) * 1.8 + 32).toFixed(0);
var val13_2 = (Number(document.getElementById('val13_2').value) * 1.8 + 32).toFixed(0);
var val14_2 = (Number(document.getElementById('val14_2').value) * 1.8 + 32).toFixed(0);
var val15_2 = (Number(document.getElementById('val15_2').value) * 1.8 + 32).toFixed(0);
var val16_2 = (Number(document.getElementById('val16_2').value) * 1.8 + 32).toFixed(0);

```

```

var par1_2 = Number(document.getElementById('par1_2').value).toFixed(0);
var par2_2 = Number(document.getElementById('par2_2').value).toFixed(0);
var kidd_2 = Number(document.getElementById('kid_2').value).toFixed(0);

```

```

var val1_5 = (Number(document.getElementById('val1_5').value) * 1.8 + 32).toFixed(0);
var val2_5 = (Number(document.getElementById('val2_5').value) * 1.8 + 32).toFixed(0);
var val3_5 = (Number(document.getElementById('val3_5').value) * 1.8 + 32).toFixed(0);
var val4_5 = (Number(document.getElementById('val4_5').value) * 1.8 + 32).toFixed(0);
var val5_5 = (Number(document.getElementById('val5_5').value) * 1.8 + 32).toFixed(0);
var val6_5 = (Number(document.getElementById('val6_5').value) * 1.8 + 32).toFixed(0);
var val7_5 = (Number(document.getElementById('val7_5').value) * 1.8 + 32).toFixed(0);
var val8_5 = (Number(document.getElementById('val8_5').value) * 1.8 + 32).toFixed(0);
var val9_5 = (Number(document.getElementById('val9_5').value) * 1.8 + 32).toFixed(0);
var val10_5 = (Number(document.getElementById('val10_5').value) * 1.8 + 32).toFixed(0);
var val11_5 = (Number(document.getElementById('val11_5').value) * 1.8 + 32).toFixed(0);
var val12_5 = (Number(document.getElementById('val12_5').value) * 1.8 + 32).toFixed(0);
var val13_5 = (Number(document.getElementById('val13_5').value) * 1.8 + 32).toFixed(0);
var val14_5 = (Number(document.getElementById('val14_5').value) * 1.8 + 32).toFixed(0);
var val15_5 = (Number(document.getElementById('val15_5').value) * 1.8 + 32).toFixed(0);
var val16_5 = (Number(document.getElementById('val16_5').value) * 1.8 + 32).toFixed(0);

```



```
var val15_20 = (Number(document.getElementById('val15_20').value) * 1.8 + 32).toFixed(0);
var val16_20 = (Number(document.getElementById('val16_20').value) * 1.8 + 32).toFixed(0);
```

```
var par1_20 = Number(document.getElementById('par1_20').value).toFixed(0);
var par2_20 = Number(document.getElementById('par2_20').value).toFixed(0);
var kidd_20 = Number(document.getElementById('kid_20').value).toFixed(0);
```

```
var val1_23 = (Number(document.getElementById('val1_23').value) * 1.8 + 32).toFixed(0);
var val2_23 = (Number(document.getElementById('val2_23').value) * 1.8 + 32).toFixed(0);
var val3_23 = (Number(document.getElementById('val3_23').value) * 1.8 + 32).toFixed(0);
var val4_23 = (Number(document.getElementById('val4_23').value) * 1.8 + 32).toFixed(0);
var val5_23 = (Number(document.getElementById('val5_23').value) * 1.8 + 32).toFixed(0);
var val6_23 = (Number(document.getElementById('val6_23').value) * 1.8 + 32).toFixed(0);
var val7_23 = (Number(document.getElementById('val7_23').value) * 1.8 + 32).toFixed(0);
var val8_23 = (Number(document.getElementById('val8_23').value) * 1.8 + 32).toFixed(0);
var val9_23 = (Number(document.getElementById('val9_23').value) * 1.8 + 32).toFixed(0);
var val10_23 = (Number(document.getElementById('val10_23').value) * 1.8 + 32).toFixed(0);
var val11_23 = (Number(document.getElementById('val11_23').value) * 1.8 + 32).toFixed(0);
var val12_23 = (Number(document.getElementById('val12_23').value) * 1.8 + 32).toFixed(0);
var val13_23 = (Number(document.getElementById('val13_23').value) * 1.8 + 32).toFixed(0);
var val14_23 = (Number(document.getElementById('val14_23').value) * 1.8 + 32).toFixed(0);
var val15_23 = (Number(document.getElementById('val15_23').value) * 1.8 + 32).toFixed(0);
var val16_23 = (Number(document.getElementById('val16_23').value) * 1.8 + 32).toFixed(0);
```

```
var par1_23 = Number(document.getElementById('par1_23').value).toFixed(0);
var par2_23 = Number(document.getElementById('par2_23').value).toFixed(0);
var kidd_23 = Number(document.getElementById('kid_23').value).toFixed(0);
```

```
function
prop_otbor(arg1,arg2,arg3,arg4,arg5,arg6,arg7,arg8,arg9,arg10,arg11,arg12,arg13,arg14,arg15,arg16){

    var arg1 = Number(arg1);
    var arg2 = Number(arg2);
    var arg3 = Number(arg3);
    var arg4 = Number(arg4);
    var arg5 = Number(arg5);
    var arg6 = Number(arg6);
    var arg7 = Number(arg7);
    var arg8 = Number(arg8);
    var arg9 = Number(arg9);
    var arg10 = Number(arg10);
    var arg11 = Number(arg11);
    var arg12 = Number(arg12);
    var arg13 = Number(arg13);
    var arg14 = Number(arg14);
    var arg15 = Number(arg15);
    var arg16 = Number(arg16);

    var sum_arg =
    (arg1+arg2+arg3+arg4+arg5+arg6+arg7+arg8+arg9+arg10+arg11+arg12+arg13+arg14+arg15+arg16)/16
    ;
```

```

var argh1 = (arg1/sum_arg).toFixed(4)*10000*16;
var argh2 = argh1 + (arg2/sum_arg).toFixed(4)*10000*15;
var   argh3 = argh2 + (arg3/sum_arg).toFixed(4)*10000*14;
var argh4 = argh3 + (arg4/sum_arg).toFixed(4)*10000*13;
var argh5 = argh4 + (arg5/sum_arg).toFixed(4)*10000*12;
var argh6 = argh5 + (arg6/sum_arg).toFixed(4)*10000*11;
var argh7 = argh6 + (arg7/sum_arg).toFixed(4)*10000*10;
var argh8 = argh7 + (arg8/sum_arg).toFixed(4)*10000*9;
var argh9 = argh8 + (arg9/sum_arg).toFixed(4)*10000*8;
var argh10 = argh9 + (arg10/sum_arg).toFixed(4)*10000*7;
var argh11 = argh10 + (arg11/sum_arg).toFixed(4)*10000*6;
var argh12 = argh11 + (arg12/sum_arg).toFixed(4)*10000*5;
var argh13 = argh12 + (arg13/sum_arg).toFixed(4)*10000*4;
var argh14 = argh13 + (arg14/sum_arg).toFixed(4)*10000*3;
var argh15 = argh14 + (arg15/sum_arg).toFixed(4)*10000*2;
var argh16 = argh15 + (arg16/sum_arg).toFixed(4)*10000*1;

```

```

var v1 = 0;
var v2 = 0;
var v3 = 0;
var v4 = 0;
var v5 = 0;
var v6 = 0;
var v7 = 0;
var v8 = 0;
var v9 = 0;
var v10 = 0;
var v11 = 0;
var v12 = 0;
var v13 = 0;
var v14 = 0;
var v15 = 0;
var v16 = 0;

```

```

var res1 = "";
var pre_res2 = "";
var res2 = "";

```

```

function rounds(kols){
for (i = 0; i < kols; i++){

```

```

    rand_argh = Math.random() * (argh16 - 1) + 1;

```

```

    if(rand_argh > 1 && rand_argh < argh1){
        v1 += 1;
    }

```

```

    if(rand_argh > argh1 && rand_argh < argh2){
        v2 += 1;
    }

```

```

    if(rand_argh > argh2 && rand_argh < argh3){

```

```
v3 += 1;
}

if(rand_argh > argh3 && rand_argh < argh4){
v4 += 1;
}

if(rand_argh > argh4 && rand_argh < argh5){
v5 += 1;
}

if(rand_argh > argh5 && rand_argh < argh6){
v6 += 1;
}

if(rand_argh > argh6 && rand_argh < argh7){
v7 += 1;
}

if(rand_argh > argh7 && rand_argh < argh8){
v8 += 1;
}

if(rand_argh > argh8 && rand_argh < argh9){
v9 += 1;
}

if(rand_argh > argh9 && rand_argh < argh10){
v10 += 1;
}

if(rand_argh > argh10 && rand_argh < argh11){
v11 += 1;
}

if(rand_argh > argh11 && rand_argh < argh12){
v12 += 1;
}

if(rand_argh > argh12 && rand_argh < argh13){
v13 += 1;
}

if(rand_argh > argh13 && rand_argh < argh14){
v14 += 1;
}

if(rand_argh > argh14 && rand_argh < argh15){
v15 += 1;
}

if(rand_argh > argh15 && rand_argh < argh16){
v16 += 1;
}
}
}
}
```

```

rounds(100);

var matches = 0;

while(matches == 0){

    if(v1 != v2 && v1 != v3 && v1 != v4 && v1 != v5 && v1 != v6 && v1 != v7 && v1 != v8
    && v1 != v9 &&
    v1 != v10 && v1 != v11 && v1 != v12 && v1 != v13 && v1 != v14 && v1 != v15 && v1 !=
    v16 &&
    v2 != v3 && v2 != v4 && v2 != v5 && v2 != v6 && v2 != v7 && v2 != v8 && v2 != v9 &&
    v2 != v10 && v2 != v11 && v2 != v12 && v2 != v13 && v2 != v14 && v2 != v15 && v2 !=
    v16 &&
    v3 != v4 && v3 != v5 && v3 != v6 && v3 != v7 && v3 != v8 && v3 != v9 && v3 != v10
    &&
    v3 != v11 && v3 != v12 && v3 != v13 && v3 != v14 && v3 != v15 && v3 != v16 &&
    v4 != v5 && v4 != v6 && v4 != v7 && v4 != v8 && v4 != v9 && v4 != v10 &&
    v4 != v11 && v4 != v12 && v4 != v13 && v4 != v14 && v4 != v15 && v4 != v16 &&
    v5 != v6 && v5 != v7 && v5 != v8 && v5 != v9 && v5 != v10 && v5 != v11 &&
    v5 != v12 && v5 != v13 && v5 != v14 && v5 != v15 && v5 != v16 &&
    v6 != v7 && v6 != v8 && v6 != v9 && v6 != v10 && v6 != v11 &&
    v6 != v12 && v6 != v13 && v6 != v14 && v6 != v15 && v6 != v16 &&
    v7 != v8 && v7 != v9 && v7 != v10 && v7 != v11 && v7 != v12 && v7 != v13 &&
    v7 != v14 && v7 != v15 && v7 != v16 &&
    v8 != v9 && v8 != v10 && v8 != v11 && v8 != v12 && v8 != v13 && v8 != v14 && v8 !=
    v15 && v8 != v16 &&
    v9 != v10 && v9 != v11 && v9 != v12 && v9 != v13 && v9 != v14 && v9 != v15 && v9 !=
    v16 &&
    v10 != v11 && v10 != v12 && v10 != v13 && v10 != v14 && v10 != v15 && v10 != v16
    &&
    v11 != v12 && v11 != v13 && v11 != v14 && v11 != v15 && v11 != v16 &&
    v12 != v13 && v12 != v14 && v12 != v15 && v12 != v16 &&
    v13 != v14 && v13 != v15 && v13 != v16 &&
    v14 != v15 && v14 != v16 &&
    v15 != v16){

        matches = 1;

        if(Math.max(v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v1){
            res1 = arg1;
            pre_res2 = Math.max(v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16);
        }

        if(Math.max(v1,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v2){
            res1 = arg2;
            pre_res2 = Math.max(v1,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16);
        }

        if(Math.max(v1,v2,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v3){
            res1 = arg3;
            pre_res2 = Math.max(v1,v2,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16);
        }

        if(Math.max(v1,v2,v3,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v4){

```

```

res1 = arg4;
pre_res2 = Math.max(v1,v2,v3,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16);
}

if(Math.max(v1,v2,v3,v4,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v5){
res1 = arg5;
pre_res2 = Math.max(v1,v2,v3,v4,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16);
}

if(Math.max(v1,v2,v3,v4,v5,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v6){
res1 = arg6;
pre_res2 = Math.max(v1,v2,v3,v4,v5,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16);
}

if(Math.max(v1,v2,v3,v4,v5,v6,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v7){
res1 = arg7;
pre_res2 = Math.max(v1,v2,v3,v4,v5,v6,v8,v9,v10,v11,v12,v13,v14,v15,v16);
}

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v9,v10,v11,v12,v13,v14,v15,v16) <= v8){
res1 = arg8;
pre_res2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v9,v10,v11,v12,v13,v14,v15,v16);
}

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v10,v11,v12,v13,v14,v15,v16) <= v9){
res1 = arg9;
pre_res2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v10,v11,v12,v13,v14,v15,v16);
}

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v11,v12,v13,v14,v15,v16) <= v10){
res1 = arg10;
pre_res2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v11,v12,v13,v14,v15,v16);
}

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v12,v13,v14,v15,v16) <= v11){
res1 = arg11;
pre_res2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v12,v13,v14,v15,v16);
}

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v13,v14,v15,v16) <= v12){
res1 = arg12;
pre_res2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v13,v14,v15,v16);
}

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v14,v15,v16) <= v13){
res1 = arg13;
pre_res2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v14,v15,v16);
}

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v15,v16) <= v14){
res1 = arg14;
pre_res2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v15,v16);
}

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v16) <= v15){
res1 = arg15;

```

```

pre_res2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v16);
}

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15) <= v16){
res1 = arg16;
pre_res2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15);
}

if(pre_res2 == v1)
res2 = arg1;

if(pre_res2 == v2)
res2 = arg2;

if(pre_res2 == v3)
res2 = arg3;

if(pre_res2 == v4)
res2 = arg4;

if(pre_res2 == v5)
res2 = arg5;

if(pre_res2 == v6)
res2 = arg6;

if(pre_res2 == v7)
res2 = arg7;

if(pre_res2 == v8)
res2 = arg8;

if(pre_res2 == v9)
res2 = arg9;

if(pre_res2 == v10)
res2 = arg10;

if(pre_res2 == v11)
res2 = arg11;

if(pre_res2 == v12)
res2 = arg12;

if(pre_res2 == v13)
res2 = arg13;

if(pre_res2 == v14)
res2 = arg14;

if(pre_res2 == v15)
res2 = arg15;

if(pre_res2 == v16)
res2 = arg16;
}else{

```

```

    rounds(20)
  }
}

var res1_prop = res1;
var res2_prop = res2;

//document.write(v1, ",",v2, ",",v3, ",",v4, ",",v5, ",",v6, ",",v7, ",",v8, ",",v9, ",",v10, ",",
,v11, ",",v12, ",",v13, ",",v14, ",",v15, ",",v16);
//alert(res1);
//alert(res2);

document.getElementById('par1').value = res1_prop;
document.getElementById('par2').value = res2_prop;
}

function
rangering(arg1,arg2,arg3,arg4,arg5,arg6,arg7,arg8,arg9,arg10,arg11,arg12,arg13,arg14,arg15,arg16){

    var arg1 = Number(arg1);
    var arg2 = Number(arg2);
    var arg3 = Number(arg3);
    var arg4 = Number(arg4);
    var arg5 = Number(arg5);
    var arg6 = Number(arg6);
    var arg7 = Number(arg7);
    var arg8 = Number(arg8);
    var arg9 = Number(arg9);
    var arg10 = Number(arg10);
    var arg11 = Number(arg11);
    var arg12 = Number(arg12);
    var arg13 = Number(arg13);
    var arg14 = Number(arg14);
    var arg15 = Number(arg15);
    var arg16 = Number(arg16);

    var alpha_max = Number((Math.random() * (2 - 1) + 1).toFixed(2));

    var N = 16;

    var argh1 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (1-1)/N-1)*100).toFixed(4);
    var argh2 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (2-1)/N-1)*100).toFixed(4);
    var argh3 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (3-1)/N-1)*100).toFixed(4);
    var argh4 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (4-1)/N-1)*100).toFixed(4);
    var argh5 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (5-1)/N-1)*100).toFixed(4);
    var argh6 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (6-1)/N-1)*100).toFixed(4);
    var argh7 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (7-1)/N-1)*100).toFixed(4);
    var argh8 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (8-1)/N-1)*100).toFixed(4);
    var argh9 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (9-1)/N-1)*100).toFixed(4);
    var argh10 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (10-1)/N-1)*100).toFixed(4);
    var argh11 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (11-1)/N-1)*100).toFixed(4);
    var argh12 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (12-1)/N-1)*100).toFixed(4);
    var argh13 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (13-1)/N-1)*100).toFixed(4);
    var argh14 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (14-1)/N-1)*100).toFixed(4);
    var argh15 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (15-1)/N-1)*100).toFixed(4);
    var argh16 = (1/N * (alpha_max - (alpha_max - 2 + alpha_max) * (16-1)/N-1)*100).toFixed(4);

```

```

var res1 = "";
var pre_res2 = "";
var res2 = "";

    if(Math.max(argh2, argh3, argh4, argh5, argh6, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14, argh15, argh16) <= argh1){
        res1 = arg1;
        pre_res2 =
Math.max(argh2, argh3, argh4, argh5, argh6, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14, argh15, argh16);
    }

    if(Math.max(argh1, argh3, argh4, argh5, argh6, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14, argh15, argh16) <= argh2){
        res1 = arg2;
        pre_res2 =
Math.max(argh1, argh3, argh4, argh5, argh6, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14, argh15, argh16);
    }

    if(Math.max(argh1, argh2, argh4, argh5, argh6, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14, argh15, argh16) <= argh3){
        res1 = arg3;
        pre_res2 =
Math.max(argh1, argh2, argh4, argh5, argh6, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14, argh15, argh16);
    }

    if(Math.max(argh1, argh2, argh3, argh5, argh6, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14, argh15, argh16) <= argh4){
        res1 = arg4;
        pre_res2 =
Math.max(argh1, argh2, argh3, argh5, argh6, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14, argh15, argh16);
    }

    if(Math.max(argh1, argh2, argh3, argh4, argh6, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14, argh15, argh16) <= argh5){
        res1 = arg5;
        pre_res2 =
Math.max(argh1, argh2, argh3, argh4, argh6, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14, argh15, argh16);
    }

    if(Math.max(argh1, argh2, argh3, argh4, argh5, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14, argh15, argh16) <= argh6){
        res1 = arg6;
        pre_res2 =
Math.max(argh1, argh2, argh3, argh4, argh5, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14, argh15, argh16);
    }

    if(Math.max(argh1, argh2, argh3, argh4, argh5, argh6, argh8, argh9, argh10, argh11, argh12, argh13, argh14, argh15, argh16) <= argh7){
        res1 = arg7;

```

```

        pre_res2 =
Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh8,argh9,argh10,argh11,argh12,argh13,argh14,argh1
5,argh16);
    }

    if(Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh7,argh9,argh10,argh11,argh12,argh13,ar
gh14,argh15,argh16) <= argh8){
        res1 = arg8;
        pre_res2 =
Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh7,argh9,argh10,argh11,argh12,argh13,argh14,argh1
5,argh16);
    }

    if(Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh7,argh8,argh10,argh11,argh12,argh13,ar
gh14,argh15,argh16) <= argh9){
        res1 = arg9;
        pre_res2 =
Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh7,argh8,argh10,argh11,argh12,argh13,argh14,argh1
5,argh16);
    }

    if(Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh7,argh8,argh9,argh11,argh12,argh13,arg
h14,argh15,argh16) <= argh10){
        res1 = arg10;
        pre_res2 =
Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh7,argh8,argh9,argh11,argh12,argh13,argh14,argh15,
argh16);
    }

    if(Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh7,argh8,argh9,argh10,argh12,argh13,arg
h14,argh15,argh16) <= argh11){
        res1 = arg11;
        pre_res2 =
Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh7,argh8,argh9,argh10,argh12,argh13,argh14,argh15,
argh16);
    }

    if(Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh7,argh8,argh9,argh10,argh11,argh13,arg
h14,argh15,argh16) <= argh12){
        res1 = arg12;
        pre_res2 =
Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh7,argh8,argh9,argh10,argh11,argh13,argh14,argh15,
argh16);
    }

    if(Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh7,argh8,argh9,argh10,argh11,argh12,arg
h14,argh15,argh16) <= argh13){
        res1 = arg13;
        pre_res2 =
Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh7,argh8,argh9,argh10,argh11,argh12,argh14,argh15,
argh16);
    }

    if(Math.max(argh1,argh2,argh3,argh4,argh5,argh6,argh7,argh8,argh9,argh10,argh11,argh12,arg
h13,argh15,argh16) <= argh14){
        res1 = arg14;

```

```

        pre_res2 =
Math.max(argh1, argh2, argh3, argh4, argh5, argh6, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh15,
argh16);
    }

    if(Math.max(argh1, argh2, argh3, argh4, argh5, argh6, argh7, argh8, argh9, argh10, argh11, argh12, arg
h13, argh14, argh16) <= argh15){
        res1 = argh15;
        pre_res2 =
Math.max(argh1, argh2, argh3, argh4, argh5, argh6, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14,
argh16);
    }

    if(Math.max(argh1, argh2, argh3, argh4, argh5, argh6, argh7, argh8, argh9, argh10, argh11, argh12, arg
h13, argh14, argh15) <= argh16){
        res1 = argh16;
        pre_res2 =
Math.max(argh1, argh2, argh3, argh4, argh5, argh6, argh7, argh8, argh9, argh10, argh11, argh12, argh13, argh14,
argh15);
    }

    if(pre_res2 == argh1)
        res2 = arg1;

    if(pre_res2 == argh2)
        res2 = arg2;

    if(pre_res2 == argh3)
        res2 = arg3;

    if(pre_res2 == argh4)
        res2 = arg4;

    if(pre_res2 == argh5)
        res2 = arg5;

    if(pre_res2 == argh6)
        res2 = arg6;

    if(pre_res2 == argh7)
        res2 = arg7;

    if(pre_res2 == argh8)
        res2 = arg8;

    if(pre_res2 == argh9)
        res2 = arg9;

    if(pre_res2 == argh10)
        res2 = arg10;

    if(pre_res2 == argh11)
        res2 = arg11;

    if(pre_res2 == argh12)
        res2 = arg12;

```

```

    if(pre_res2 == argh13)
    res2 = arg13;

    if(pre_res2 == argh14)
    res2 = arg14;

    if(pre_res2 == argh15)
    res2 = arg15;

    if(pre_res2 == argh16)
    res2 = arg16;

    /*document.write(argh1, "," ,argh2, "," ,argh3, "," ,argh4, "," ,argh5, "," ,argh6, "," ,argh7, ","
,argh8, "," ,argh9, "," ,argh10, "," ,argh11, "," ,argh12, "," ,argh13, "," ,argh14, "," ,argh15, "," ,argh16);
    alert(res1);
    alert(res2);*/

    var res1_rang = res1;
    var res2_rang = res2;

    document.getElementById('par1').value = res1_rang;
    document.getElementById('par2').value = res2_rang;
    //d();

}

function
turnir_otbor(arg1,arg2,arg3,arg4,arg5,arg6,arg7,arg8,arg9,arg10,arg11,arg12,arg13,arg14,arg15,arg16){

    var arg1 = Number(arg1);
    var arg2 = Number(arg2);
    var arg3 = Number(arg3);
    var arg4 = Number(arg4);
    var arg5 = Number(arg5);
    var arg6 = Number(arg6);
    var arg7 = Number(arg7);
    var arg8 = Number(arg8);
    var arg9 = Number(arg9);
    var arg10 = Number(arg10);
    var arg11 = Number(arg11);
    var arg12 = Number(arg12);
    var arg13 = Number(arg13);
    var arg14 = Number(arg14);
    var arg15 = Number(arg15);
    var arg16 = Number(arg16);

    var sum_arg =
    (arg1+arg2+arg3+arg4+arg5+arg6+arg7+arg8+arg9+arg10+arg11+arg12+arg13+arg14+arg15+arg16)/16
    ;

    var argh1 = (arg1/sum_arg).toFixed(4);
    var argh2 = (arg2/sum_arg).toFixed(4);
    var   argh3 = (arg3/sum_arg).toFixed(4);

```

```

var argh4 = (arg4/sum_arg).toFixed(4);
var argh5 = (arg5/sum_arg).toFixed(4);
var argh6 = (arg6/sum_arg).toFixed(4);
var argh7 = (arg7/sum_arg).toFixed(4);
var argh8 = (arg8/sum_arg).toFixed(4);
var argh9 = (arg9/sum_arg).toFixed(4);
var argh10 = (arg10/sum_arg).toFixed(4);
var argh11 = (arg11/sum_arg).toFixed(4);
var argh12 = (arg12/sum_arg).toFixed(4);
var argh13 = (arg13/sum_arg).toFixed(4);
var argh14 = (arg14/sum_arg).toFixed(4);
var argh15 = (arg15/sum_arg).toFixed(4);
var argh16 = (arg16/sum_arg).toFixed(4);

```

```

var res1 = "";
var res2 = "";
var res3 = "";
var res4 = "";
var res5 = "";
var res6 = "";
var res7 = "";
var res8 = "";

```

```

var res_s_f1 = "";
var res_s_f2 = "";
var res_s_f3 = "";
var res_s_f4 = "";

```

```

var res_f1 = "";
var res_f2 = "";

```

```

if (argh1 >= argh5){

    res1 = argh1;

    }else{
        res1 = argh2;
    }

```

```

if (argh2 >= argh6){

    res2 = argh2;

    }else{

        res2 = argh6;

    }

```

```

if (argh3 >= argh7){

    res3 = argh3;

```

```
    }else{

        res3 = argh7;

    }

    if (argh4 >= argh8){

        res4 = argh4;

    }else{

        res4 = argh8;

    }

    if (argh9 >= argh13){

        res5 = argh9;

    }else{

        res5 = argh13;

    }

    if (argh10 >= argh14){

        res6 = argh10;

    }else{

        res6 = argh14;

    }

    if (argh11 >= argh15){

        res7 = argh11;

    }else{

        res7 = argh11;

    }

    if (argh12 >= argh16){

        res8 = argh12;
```

```
    }else{

        res8 = argh16;

    }

if (res1 >= res5){

    res_s_f1 = res1;

    }else{

        res_s_f1 = res5;

    }

if (res2 >= res6){

    res_s_f2 = res2;

    }else{

        res_s_f2 = res6;

    }

if (res3 >= res7){

    res_s_f3 = res3;

    }else{

        res_s_f3 = res7;

    }

if (res4 >= res8){

    res_s_f4 = res4;

    }else{

        res_s_f4 = res8;

    }
```

```
if (res_s_f1 >= res_s_f3){  
  
    res_f1 = res_s_f1;  
  
}else{  
  
    res_f1 = res_s_f3;  
  
}  
  
if (res_s_f2 >= res_s_f4){  
  
    res_f2 = res_s_f2;  
  
}else{  
  
    res_f2 = res_s_f4;  
  
}
```

```
if(res_f1 == argh1)  
res_f1 = arg1;  
  
if(res_f1 == argh2)  
res_f1 = arg2;  
  
if(res_f1 == argh3)  
res_f1 = arg3;  
  
if(res_f1 == argh4)  
res_f1 = arg4;  
  
if(res_f1 == argh5)  
res_f1 = arg5;  
  
if(res_f1 == argh6)  
res_f1 = arg6;  
  
if(res_f1 == argh7)  
res_f1 = arg7;  
  
if(res_f1 == argh8)  
res_f1 = arg8;  
  
if(res_f1 == argh9)  
res_f1 = arg9;  
  
if(res_f1 == argh10)  
res_f1 = arg10;
```

```
if(res_f1 == argh11)
res_f1 = arg11;
```

```
if(res_f1 == argh12)
res_f1 = arg12;
```

```
if(res_f1 == argh13)
res_f1 = arg13;
```

```
if(res_f1 == argh14)
res_f1 = arg14;
```

```
if(res_f1 == argh15)
res_f1 = arg15;
```

```
if(res_f1 == argh16)
res_f1 = arg16;
```

```
if(res_f2 == argh1)
res_f2 = arg1;
```

```
if(res_f2 == argh2)
res_f2 = arg2;
```

```
if(res_f2 == argh3)
res_f2 = arg3;
```

```
if(res_f2 == argh4)
res_f2 = arg4;
```

```
if(res_f2 == argh5)
res_f2 = arg5;
```

```
if(res_f2 == argh6)
res_f2 = arg6;
```

```
if(res_f2 == argh7)
res_f2 = arg7;
```

```
if(res_f2 == argh8)
res_f2 = arg8;
```

```
if(res_f2 == argh9)
res_f2 = arg9;
```

```
if(res_f2 == argh10)
res_f2 = arg10;
```

```
if(res_f2 == argh11)
res_f2 = arg11;
```

```
if(res_f2 == argh12)
res_f2 = arg12;
```

```

    if(res_f2 == argh13)
    res_f2 = arg13;

    if(res_f2 == argh14)
    res_f2 = arg14;

    if(res_f2 == argh15)
    res_f2 = arg15;

    if(res_f2 == argh16)
    res_f2 = arg16;


    //alert(res_f1);
    //alert(res_f2);

    var res1_turn = res_f1;
    var res2_turn = res_f2;


    //document.write(argh1, "," ,argh2, "," ,argh3, "," ,argh4, "," ,argh5, "," ,argh6, "," ,argh7, ","
,argh8, "," ,argh9, "," ,argh10, "," ,argh11, "," ,argh12, "," ,argh13, "," ,argh14, "," ,argh15, "," ,argh16);
    document.getElementById('par1').value = res1_turn;
    document.getElementById('par2').value = res2_turn;
    //d();

}

function
porog_otbor(arg1,arg2,arg3,arg4,arg5,arg6,arg7,arg8,arg9,arg10,arg11,arg12,arg13,arg14,arg15,arg16){

    var arg1 = Number(arg1);
    var arg2 = Number(arg2);
    var arg3 = Number(arg3);
    var arg4 = Number(arg4);
    var arg5 = Number(arg5);
    var arg6 = Number(arg6);
    var arg7 = Number(arg7);
    var arg8 = Number(arg8);
    var arg9 = Number(arg9);
    var arg10 = Number(arg10);
    var arg11 = Number(arg11);
    var arg12 = Number(arg12);
    var arg13 = Number(arg13);
    var arg14 = Number(arg14);
    var arg15 = Number(arg15);
    var arg16 = Number(arg16);

    var sum_arg =
    (arg1+arg2+arg3+arg4+arg5+arg6+arg7+arg8+arg9+arg10+arg11+arg12+arg13+arg14+arg15+arg16)/16
    ;

    var argh1 = (arg1/sum_arg).toFixed(4);

```

```

var argh2 = (arg2/sum_arg).toFixed(4);
var argh3 = (arg3/sum_arg).toFixed(4);
var argh4 = (arg4/sum_arg).toFixed(4);
var argh5 = (arg5/sum_arg).toFixed(4);
var argh6 = (arg6/sum_arg).toFixed(4);
var argh7 = (arg7/sum_arg).toFixed(4);
var argh8 = (arg8/sum_arg).toFixed(4);
var argh9 = (arg9/sum_arg).toFixed(4);
var argh10 = (arg10/sum_arg).toFixed(4);
var argh11 = (arg11/sum_arg).toFixed(4);
var argh12 = (arg12/sum_arg).toFixed(4);
var argh13 = (arg13/sum_arg).toFixed(4);
var argh14 = (arg14/sum_arg).toFixed(4);
var argh15 = (arg15/sum_arg).toFixed(4);
var argh16 = (arg16/sum_arg).toFixed(4);

```

```

var res1 = 0;
var res2 = 0;
var res3 = 0;
var res4 = 0;
var res5 = 0;
var res6 = 0;
var res7 = 0;
var res8 = 0;
var res9 = 0;
var res10 = 0;
var res11 = 0;
var res12 = 0;
var res13 = 0;
var res14 = 0;
var res15 = 0;
var res16 = 0;

```

```

if (argh1 >= 1){

    res1 = argh1;

```

```

}

```

```

if (argh2 >= 1){

    res2 = argh2;

```

```

}

```

```

if (argh3 >= 1){

    res3 = argh3;

```

```

}

```

```

if (argh4 >= 1){

```

```
        res4 = argh4;
    }
    if (argh5 >= 1){
        res5 = argh5;
    }
    if (argh6 >= 1){
        res6 = argh6;
    }
    if (argh7 >= 1){
        res7 = argh7;
    }
    if (argh8 >= 1){
        res8 = argh8;
    }
    if (argh9 >= 1){
        res9 = argh9;
    }
    if (argh10 >= 1){
        res10 = argh10;
    }
    if (argh11 >= 1){
        res11 = argh11;
    }
    if (argh12 >= 1){
        res12 = argh12;
```

```
}  
  
if (argh13 >= 1){  
  
    res13 = argh13;  
  
}  
  
if (argh14 >= 1){  
  
    res14 = argh14;  
  
}  
  
if (argh15 >= 1){  
  
    res15 = argh15;  
  
}  
  
if (argh16 >= 1){  
  
    res16 = argh16;  
  
}
```

```
var v1 = 0;  
var v2 = 0;  
var v3 = 0;  
var v4 = 0;  
var v5 = 0;  
var v6 = 0;  
var v7 = 0;  
var v8 = 0;  
var v9 = 0;  
var v10 = 0;  
var v11 = 0;  
var v12 = 0;  
var v13 = 0;  
var v14 = 0;  
var v15 = 0;  
var v16 = 0;
```

```
if (res1 != ""){  
  
v1 = 1;  
  
}
```

```
if (res2 != ""){  
  
v2 = 1;  
  
}  
  
if (res3 != ""){  
  
v3 = 1;  
  
}  
  
if (res4 != ""){  
  
v4 = 1;  
  
}  
  
if (res5 != ""){  
  
v5 = 1;  
  
}  
  
if (res6 != ""){  
  
v6 = 1;  
  
}  
  
if (res7 != ""){  
  
v7 = 1;  
  
}  
  
if (res8 != ""){  
  
v8 = 1;  
  
}  
  
if (res9 != ""){  
  
v9 = 1;  
  
}  
  
if (res10 != ""){
```

```
v10 = 1;

}

if (res11 != ""){

v11 = 1;

}

if (res12 != ""){

v12 = 1;

}

if (res13 != ""){

v13 = 1;

}

if (res14 != ""){

v14 = 1;

}

if (res15 != ""){

v15 = 1;

}

if (res16 != ""){

v16 = 1;

}


function rounds(kols){

    for (i = 0; i < kols; i++){

rand_argh = (Math.random() * (16 - 1) + 1).toFixed(0);

if(rand_argh == 1 && v1 != 0){

v1 += 1;

}
```

```
if(rand_argh == 2 && v2 != 0){  
  
    v2 += 1;  
  
}  
  
if(rand_argh == 3 && v3 != 0){  
  
    v3 += 1;  
  
}  
  
if(rand_argh == 4 && v4 != 0){  
  
    v4 += 1;  
  
}  
  
if(rand_argh == 5 && v5 != 0){  
  
    v5 += 1;  
  
}  
  
if(rand_argh == 6 && v6 != 0){  
  
    v6 += 1;  
  
}  
  
if(rand_argh == 7 && v7 != 0){  
  
    v7 += 1;  
  
}  
  
if(rand_argh == 8 && v8 != 0){  
  
    v8 += 1;  
  
}  
  
if(rand_argh == 9 && v9 != 0){  
  
    v9 += 1;  
  
}  
  
if(rand_argh == 10 && v10 != 0){  
  
    v10 += 1;  
  
}
```

```

if(rand_argh == 11 && v11 != 0){

v11 += 1;

}

if(rand_argh == 12 && v12 != 0){

v12 += 1;

}

if(rand_argh == 13 && v13 != 0){

v13 += 1;

}

if(rand_argh == 14 && v14 != 0){

v14 += 1;

}

if(rand_argh == 15 && v15 != 0){

v15 += 1;

}

if(rand_argh == 16 && v16 != 0){

v16 += 1;

}

}
}
}

rounds(10000);

var matches = 0;

result1 = "";
result2 = "";
pre_result2 = "";

if(Math.max(v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v1){
result1 = arg1;
pre_result2 = Math.max(v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16);
}

```

```

if(Math.max(v1,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v2){
result1 = arg2;
pre_result2 = Math.max(v1,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16);
}

```

```

if(Math.max(v1,v2,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v3){
result1 = arg3;
pre_result2 = Math.max(v1,v2,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16);
}

```

```

if(Math.max(v1,v2,v3,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v4){
result1 = arg4;
pre_result2 = Math.max(v1,v2,v3,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16);
}

```

```

if(Math.max(v1,v2,v3,v4,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v5){
result1 = arg5;
pre_result2 = Math.max(v1,v2,v3,v4,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16);
}

```

```

if(Math.max(v1,v2,v3,v4,v5,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v6){
result1 = arg6;
pre_result2 = Math.max(v1,v2,v3,v4,v5,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16);
}

```

```

if(Math.max(v1,v2,v3,v4,v5,v6,v8,v9,v10,v11,v12,v13,v14,v15,v16) <= v7){
result1 = arg7;
pre_result2 = Math.max(v1,v2,v3,v4,v5,v6,v8,v9,v10,v11,v12,v13,v14,v15,v16);
}

```

```

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v9,v10,v11,v12,v13,v14,v15,v16) <= v8){
result1 = arg8;
pre_result2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v9,v10,v11,v12,v13,v14,v15,v16);
}

```

```

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v10,v11,v12,v13,v14,v15,v16) <= v9){
result1 = arg9;
pre_result2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v10,v11,v12,v13,v14,v15,v16);
}

```

```

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v11,v12,v13,v14,v15,v16) <= v10){
result1 = arg10;
pre_result2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v11,v12,v13,v14,v15,v16);
}

```

```

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v12,v13,v14,v15,v16) <= v11){
result1 = arg11;
pre_result2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v12,v13,v14,v15,v16);
}

```

```

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v13,v14,v15,v16) <= v12){
result1 = arg12;
pre_result2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v13,v14,v15,v16);
}

```

```

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v14,v15,v16) <= v13){
result1 = arg13;
pre_result2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v14,v15,v16);
}

```

```

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v15,v16) <= v14){
result1 = arg14;
pre_result2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v15,v16);
}

```

```

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v16) <= v15){
result1 = arg15;
pre_result2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v16);
}

```

```

if(Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15) <= v16){
result1 = arg16;
pre_result2 = Math.max(v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15);
}

```

```

if(pre_result2 == v1)
result2 = arg1;

```

```

if(pre_result2 == v2)
result2 = arg2;

```

```

if(pre_result2 == v3)
result2 = arg3;

```

```

if(pre_result2 == v4)
result2 = arg4;

```

```

if(pre_result2 == v5)
result2 = arg5;

```

```

if(pre_result2 == v6)
result2 = arg6;

```

```

if(pre_result2 == v7)
result2 = arg7;

```

```

if(pre_result2 == v8)
result2 = arg8;

```

```

if(pre_result2 == v9)
result2 = arg9;

```

```

if(pre_result2 == v10)
result2 = arg10;

```

```

if(pre_result2 == v11)
result2 = arg11;

```

```

if(pre_result2 == v12)
result2 = arg12;

```

```

if(pre_result2 == v13)
result2 = arg13;

if(pre_result2 == v14)
result2 = arg14;

if(pre_result2 == v15)
result2 = arg15;

if(pre_result2 == v16)
result2 = arg16;

/*document.write(v1, "," ,v2, "," ,v3, "," ,v4, "," ,v5, "," ,v6, "," ,v7, "," ,v8, "," ,v9, "," ,v10, "," ,
,v11, "," ,v12, "," ,v13, "," ,v14, "," ,v15, "," ,v16);
alert(result1);
alert(result2);*/

var res1_por = result1;
var res2_por = result2;

document.getElementById('par1').value = res1_por;
document.getElementById('par2').value = res2_por;
//d();
}

function chrest_odn(arg1,arg2){

arg1 = Number(arg1);
arg2 = Number(arg2);
var argh1 = arg1.toString(2);
var argh2 = arg2.toString(2);

if(argh1.length > argh2.length){
var argh_max = argh1.length;
}
else{
var argh_max = argh2.length;
}

var kid = "";

for (i = 1; i < argh_max; i++){

rand_argh = (Math.random() * (1 - 0) + 0).toFixed(0);

if (rand_argh == 1){

kid += argh1.charAt(i);

}

if (rand_argh == 0){

```

```

    kid += argh2.charAt(i);

}

}

kid = parseInt( kid, 2 );

//alert(kid);

var kid_odn = kid;

document.getElementById('kid').value = kid_odn;
//d();

}

function chrest_ravnom(arg1,arg2){

    arg1 = Number(arg1);
    arg2 = Number(arg2);
    var argh1 = arg1.toString(2);
    var argh2 = arg2.toString(2);

    if(argh1.length > argh2.length){
        var argh_max = argh1.length;
    }
    else{
        var argh_max = argh2.length;
    }

    var mask = "";
    var kid = "";

    for (i = 1; i < argh_max; i++){

        rand_argh = (Math.random() * (1 - 0) + 0).toFixed(0);

        mask += rand_argh;

    }

    for (i = 0; i < mask.length; i++){

        if(mask.charAt(i) == 0)
            kid += argh1.charAt(i);

        if(mask.charAt(i) == 1)
            kid += argh2.charAt(i);
    }

```

```

}

kid = parseInt( kid, 2 );

kid_ravnom = kid;

document.getElementById('kid').value = kid_ravnom;

}

function chrest_sravv(arg1,arg2){

arg1 = Number(arg1);
arg2 = Number(arg2);
var argh1 = arg1.toString(2);
var argh2 = arg2.toString(2);

if(argh1.length > argh2.length){
var argh_max = argh1.length;
}
else{
var argh_max = argh2.length;
}

var kid = "";

for(i = 1; i < argh_max; i++){

if(argh1.charAt(i) == argh2.charAt(i)){

kid += argh1.charAt(i);

}
else{

rand_argh = (Math.random() * (1 - 0) + 0).toFixed(0);
if (rand_argh == 0){

kid += argh1.charAt(i);

}
else{

kid += argh2.charAt(i);

}

}

}

}

while(kid.length < argh_max){
var k = argh_max - kid.length;

```

```

        var first = "0" * k;
        kid = first + kid;
    }
    alert(kid);

    kid = parseInt( kid, 2 );

    var kid_sravn = kid;
    /*ps=' forum';
    s="cyber .ru";
    ss=s.substring(0, 5);
    sss=s.substring(5);
    alert(ss+ps+sss);*/

    document.getElementById('kid').value = kid_sravn;

}

function chrest_arifm(arg1,arg2){

    arg1 = Number(arg1);
    arg2 = Number(arg2);
    //var argh1 = arg1.toString(2);
    //var argh2 = arg2.toString(2);

    var kid = "";
    var koef = 0;

    koef = Math.abs(arg1-arg2);

    kid = arg1 + koef * (arg1 - arg2);

    //kid = parseInt( kid, 2 );

    var kid_arifm = kid;

    document.getElementById('kid').value = kid_arifm;

}

function mut_prost(arg1){

    arg1 = Number(arg1);
    argh1 = arg1.toString(2);
    var dl = argh1.length;
    rand_argh = (Math.random() * (dl - 0) + 0).toFixed(0);
    var res = "";
    var shans = (Math.random() * (1 - 0) + 0).toFixed(2);

    var argh1_p1 = argh1.substring(0, rand_argh-1);
    var argh1_p2 = argh1.substring(rand_argh);

```

```

if(shans<1/16){
if(argh1.charAt(rand_argh) == "1"){

    res = argh1_p1 + "0" + argh1_p2;

}

if(argh1.charAt(rand_argh) == "0"){

    res = argh1_p1 + "1" + argh1_p2;

}
} else{
    res = argh1;
}

res = parseInt( res, 2 );

var kid_m_prost = res;

document.getElementById('kid').value = kid_m_prost;

}

function mut_tochk(arg1){

arg1 = Number(arg1);
argh1 = arg1.toString(2);
var dl = argh1.length;
var res = "";

//var argh1_p1 = argh1.substring(0, rand_argh-1);
//var argh1_p2 = argh1.substring(rand_argh);

for(i = 1;i < dl; i++){
    var shans = (Math.random() * (1 - 0) + 0).toFixed(2);
    var argh1_p1 = argh1.substring(0, rand_argh-1);
    var argh1_p2 = argh1.substring(rand_argh);

    if(shans < 1/16){

        if(argh1.charAt(i) == "0"){

            res += "1";

        }

        if(argh1.charAt(i) == "1"){

```

```

        res += "0";

    }

    }
    else{
        res += argh1.charAt(i);
    }
}

res = parseInt( res, 2 );

var kid_m_tochk = res;

document.getElementById('kid').value = kid_m_tochk;

}

function mut_sluch(arg1){

arg1 = Number(arg1);
argh1 = arg1.toString(2);
var dl = argh1.length;
var res = "";

//var argh1_p1 = argh1.substring(0, rand_argh-1);
//var argh1_p2 = argh1.substring(rand_argh);

for(i = 0;i < dl; i++){

    var qt = (Math.log(dl) - Math.log(i))/Math.log(dl);
    var r = (Math.random() * (1 - 0) + 0).toFixed(2);
    var gen = (Math.random() * (r*qt - 0) + 0).toFixed(2);
    var porg = (Math.random() * (1 - 0) + 0).toFixed(2);

    if(gen >= porg){

        if(argh1.charAt(i) == "0")
            res += "1";

        if(argh1.charAt(i) == "1")
            res += "0";

    }else{

        res += argh1.charAt(i);

    }
}

```

```

    }

    res = parseInt(res, 2);

    var kid_m_sluch = res;

    document.getElementById('kid').value = kid_m_sluch;

    }

    function mut_gauss(arg1){

        arg1 = Number(arg1);
        //argh1 = arg1.toString(2);
        //var dl = argh1.length;
        var res = "";
        var sr =
(Math.max(val1,val2,val3,val4,val5,val6,val7,val8,val9,val10,val11,val12,val13,val14,val15,val16) -
    Math.min(val1,val2,val3,val4,val5,val6,val7,val8,val9,val10,val11,val12,val13,val14,val15,val1
6))/2;

        var sigma = (15/16) *
((Math.max(val1,val2,val3,val4,val5,val6,val7,val8,val9,val10,val11,val12,val13,val14,val15,val16) -
    Math.min(val1,val2,val3,val4,val5,val6,val7,val8,val9,val10,val11,val12,val13,val14,val15,val1
6))/3);

        //alert(sigma);

        var func = (1/(sigma * Math.sqrt(2 * Math.PI))) * Math.pow(Math.exp(1,-1 * Math.pow((arg1
- sr),2)/(2 * Math.pow(sigma,2))));

        //alert(Math.pow(sigma,2));
        //alert(func);

        res = (arg1 + func).toFixed(0);

        var kid_m_gauss = res;

        //alert(res);

        document.getElementById('kid').value = kid_m_gauss;

    }

```

ДОДАТОК Б

Слайди презентації

Національний університет «Запорізька політехніка»

Кафедра програмних засобів

Дипломна кваліфікаційна робота бакалавра

Розробка інформаційного застосунку з генетичними алгоритмами для прогнозу погоди

Виконав
ст. гр. КНТ-229

Данило ЖИХАРЄВ

Керівник
к.т.н., доцент

Ірина ЗЕЛЕНЬОВА

1

Рисунок Б.1 – Слайд 1

Об'єкт дослідження – процес розробки програмного забезпечення для прогнозу погоди.

Об'єкт розробки – розробка інформаційного застосунку з генетичними алгоритмами для прогнозу погоди (Information App Development with Genetic Algorithms for Weather Forecasting).

Предмет дослідження – програмні засоби для прогнозу погоди.

Мета роботи – розробка програмного засобу, здатного прогнозувати погоду для певного регіону, побудованого на базі машинного навчання нейронних мереж генетичними алгоритмами.

2

Рисунок Б.2 – Слайд 2

ПРОГРАМНІ ЗАСОБИ ДЛЯ ПРОГНОЗУ ПОГОДИ



Рисунок 1.1 – Інформаційна система meteo.ua [5]

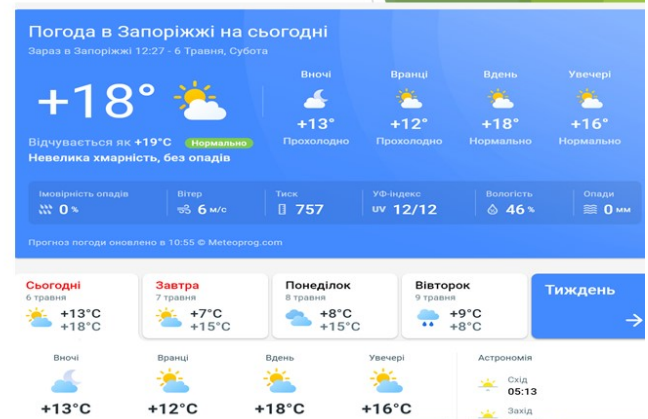


Рисунок 1.2 – Інформаційна система meteoproг [6]

Рисунок Б.3 – Слайд 3

ПРОГРАМНІ ЗАСОБИ ДЛЯ ПРОГНОЗУ ПОГОДИ

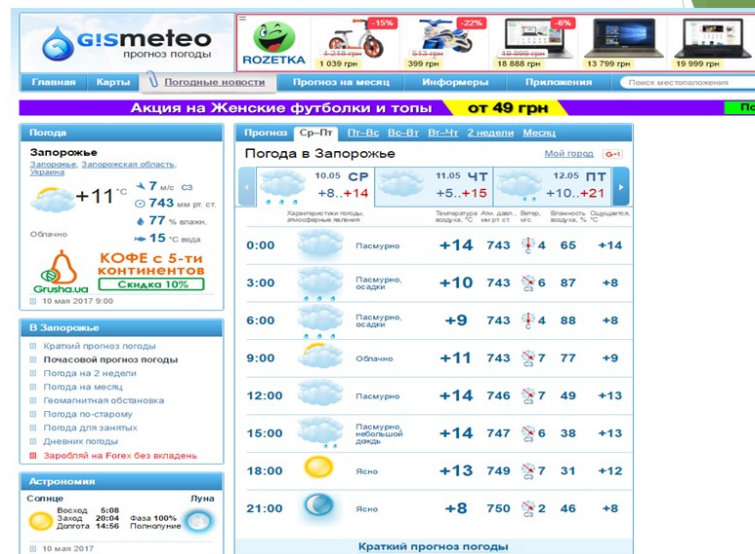


Рисунок 1.3 – Інформаційна система Gismeteo [7]

Рисунок Б.4 – Слайд 4

ПОРІВНЯННЯ АЛГОРИТМІВ

Метод	Переваги	Недоліки
1	2	3
Чисельна модель прогнозування	Легко розуміти та реалізовувати.	Не бере до уваги тенденції та сезонність ряду, тому не дуже точний для прогнозування довгострокових тенденцій або змін в сезонності.
	Швидко вираховує прогнози та використовує мало обчислювальних ресурсів.	Дуже чутливий до випадкових змін, тобто враховує вагу кожного спостереження однаково, що може призводити до неточних прогнозів при виникненні великих випадкових змін.
2 Модель прогнозування центру European Centre for medium-range weather forecasts	Легко розуміти та реалізовувати.	Не бере до уваги тенденції та сезонність ряду, тому не може бути використаний для прогнозування рядів з сезонністю або трендом.

1	2	3
□	Показує кращі результати в порівнянні з чисельною моделлю прогнозування в більш складних рядах.	Не ефективний для рядів, де інтервали часу різні, наприклад, для даних з місячною та щоденною частотою.
Модель Weather Research and Forecasting (WRF)	Бере до уваги тенденції та сезонність ряду, що дозволяє більш точно прогнозувати дані.	Не рекомендується для рядів зі складними тенденціями та сезонністю, так як цей метод не може обробляти більш складні ряди.
	Легко налаштовується для різних характеристик ряду, що дозволяє працювати з різними типами даних.	Чутливий до випадкових змін в даних, що може призвести до неприродних змін у прогнозах.
Модель Навчання штучних нейронних мереж генетичними алгоритмами	Бере до уваги тенденції та сезонність ряду, що дозволяє більш точно прогнозувати дані.	Вимагає великої кількості даних для належного підбору параметрів.
	Легко налаштовується та підтримує автоматичне оновлення параметрів, що зменшує витрати часу та зусиль.	Не може виявляти зміну в режимі роботи, якщо тенденція та сезонність ряду змінюються швидко.

5

Рисунок Б.5 – Слайд 5

ОБҐРУНТУВАННЯ ВИБОРУ МОВИ ПРОГРАМУВАННЯ

Критерій порівняння мов програмування	Мова програмування		
	CSP	JavaScript	HTML + CSS
Вебінтерфейс	+	+	+
Управління сеансами	+	—	—
Швидкість виконання коду	+	+—	+—
Багатопарадигмальність	+	+—	+
Кросплатформеність розробки	+	+—	+
Зручність використання	+	+—	+—

6

Рисунок Б.6 – Слайд 6

ER ДІАГРАМА ПРОГРАМИ



Рисунок Б.7 – Слайд 7

СХЕМА АЛГОРИТМУ ФОРМУВАННЯ РЕЗУЛЬТАТУ НА ЗАПИТ КОРИСТУВАЧА

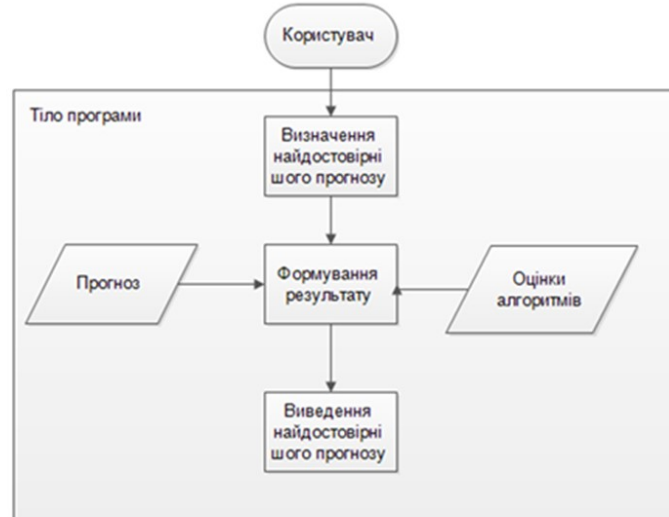
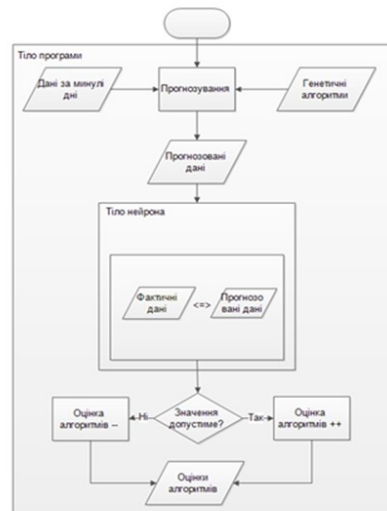


Рисунок Б.8 – Слайд 8

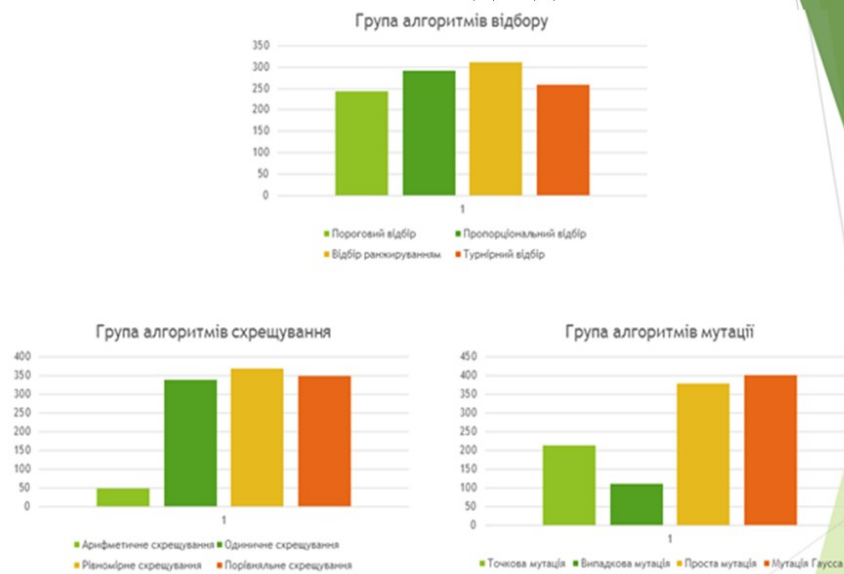
СХЕМА АЛГОРИТМУ ФОРМУВАННЯ ПРОГНОЗІВ ТА ОЦІНЮВАННЯ РОБОТИ АЛГОРИТМІВ ПРОГНОЗУВАННЯ



9

Рисунок Б.9 – Слайд 9

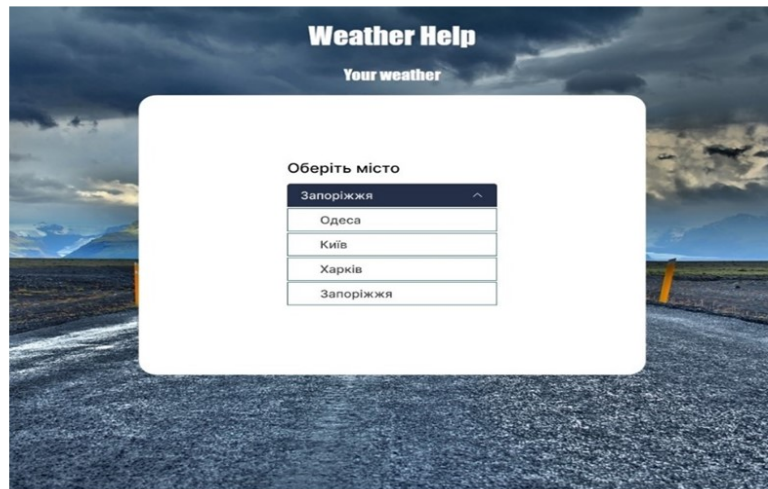
ПОРІВНЯННЯ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ ДЛЯ ПРОГНОЗУВАННЯ ПОГОДИ



10

Рисунок Б.10 – Слайд 10

СТОРІНКА ВИБОРУ МІСТА



11

Рисунок Б.11 – Слайд 11

СТОРІНКА «ПОГОДА»

Day	Time	T °C	Press	Hum	WD	WS	CI %
26.01.2023	2.00	-5	763	50	North	3	90-100
26.01.2023	5.00	-4	763	50	North	0	90-100
26.01.2023	8.00	-2	765	60	North	2	90-100
26.01.2023	11.00	0	766	90	North	0	70-80
26.01.2023	14.00	0	767	100	West	3	90-100
26.01.2023	17.00	2	766	100	North	0	90-100
26.01.2023	20.00	0	764	90	North	0	90-100
26.01.2023	23.00	-4	763	90	North-West	0	90-100

12

Рисунок Б.12 – Слайд 12

ВИСНОВКИ

Розроблена програма відповідає усім основним вимогам для ефективної роботи інформаційного застосунку «Прогноз погоди». Реалізована нейронна мережа, навчена генетичними алгоритмами, являє собою приклад елементарного штучного інтелекту, здатного до самонавчання. Пропрацювавши деякий час, система видала результати ефективності кожного алгоритма з групи алгоритмів відбору, алгоритмів схрещування, та алгоритмів мутації, що дозволило визначити який був найефективнішим у своїй групі.

Отримані результати демонструють нам ефективність обраної нами нейронної мережі, тому кожного разу перед тим як продемонструвати користувачу прогноз на новий день, програма буде обирати прогноз, у якому були задіяні найефективніші представники своєї групи.

Були досягнені наступні цілі:

- проаналізовано аналогічні системи засновані на машинному навчанні нейронних мереж;
- вивчено архітектуру та можливості середовища Tuscache;
- розроблено методи, які засновані на можливостях і особливостях генетичних алгоритмів та інтерпретовано їх під машинний код;
- розроблено методи, які засновані на можливостях і особливостях нейронних мереж та інтерпретовано їх під машинний код;
- навчено нейронні мережі генетичними алгоритмами;
- продемонстровано можливості розробленого продукту на прикладі прогнозування погоди.

Проведені експерименти засвідчили, що програма має достатню працездатність і може використовуватися за призначенням для прогнозування погоди.

Рисунок Б.13 – Слайд 13