

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Інформаційна технологія оцінки функціонального стану операторів інформаційних систем»

за напрямом підготовки 6.050101 «Комп'ютерні науки»

Виконавець роботи: студент групи ІТ-51 Вакал Світлана Миколаївна

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ «__» _____ 2019 р.

Науковий керівник

(підпис)

д.т.н., професор, Лавров Є. А.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Напрямок підготовки – 6.050101 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
«__» _____ 2019 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Вакал Світлана Миколаївна

1 Тема роботи Інформаційна технологія оцінки функціонального стану операторів інформаційних систем

керівник роботи Лавров Євгеній Анатолійович, д.т.н., професор,

затверджені наказом по університету від «17» травня 2019 р. № 084-III

2 Строк подання студентом роботи «3» червня 2019 р.

3 Вхідні дані до роботи літературні джерела з питань оцінки функціонального стану операторів, експериментальні дослідження функціонального стану операторів, література з питань нейромережевого моделювання.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз методів і засобів оцінки функціонального стану операторів автоматизованих технологічних комплексів; Постановка задачі; Проектування інформаційної системи оцінки функціонального стану операторів інформаційних систем; Розробка інформаційної системи оцінки функціонального стану операторів

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Актуальність, Постановка задачі, Аналіз предметної області, Модель функціональної мережі, Структура програмного забезпечення; Результати експериментальних досліджень, Акти впровадження, Апробація, Висновки.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____ 3.09.2018

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Визначення предметної області. Підбір літератури.	До 2.10.18	
2	Аналіз методів і засобів оцінки функціонального стану (ФС) операторів АТК	До 20.12.18	
3	Вибір засобів реалізації	До 11.01.19	
4	Проектування інформаційної системи оцінки ФС операторів інформаційних систем	До 15.02.19	
5	Розробка математичної моделі для оцінки ФС операторів	До 1.03.19	
6	Розробка інформаційної системи по визначенню ФС оператора	До 10.04.19	
7	Проведення експериментальних досліджень	До 19.04.19	
8	Оформлення пояснювальної записки (ПЗ)	До 30.05.19	
9	Здача ПЗ керівнику	До 1.06.19	
10	Виправлення помилок і перевірка записки на плагіат	До 7.06.19	
11	Здача ПЗ на кафедрі	До 10.06.19	

Студент _____

(підпис)

Вакал С. М.

Керівник роботи _____

(підпис)

д.т.н., професор Лавров Є.А.

РЕФЕРАТ

Тема роботи «Інформаційна технологія оцінки функціонального стану операторів інформаційних систем».

Мета роботи: розробити систему для оцінки функціонального стану операторів інформаційних систем.

Робота над дипломним проектом містить повторення та вивчення функціональних можливостей мови програмування Microsoft Visual Basic, пакету прикладних програм Matlab, табличного процесору Microsoft Excel та текстових редакторів з розширенням .txt. Використання перелічених програмних засобів в комплексі вирішило задачу кластеризації функціонального стану операторів інформаційних систем.

За результатами проведеного аналізу методів оцінки функціонального стану виділено метод аналізу клавіатурного почерку як найбільш оптимальний.

У результаті проведеної дипломної роботи створено програму, яка підлаштовується під функціональні особливості кожного оператора. Система збирає введені користувачем дані до текстових файлів і передає до новоствореної нейронної мережі, де відбувається її навчання. Наступний крок – введення одного з ключових слів у хаотичному порядку та передача їх даних до вже існуючої нейронної мережі, де порівнюються нові значення з еталонними, що видала попередньо мережа. Фінальним є виведення повідомлення про поточну оцінку функціонального стану операторів інформаційних систем та запис результату до файлу з статистичними даними.

Дипломний проект містить 128 сторінок, 12 таблиць, 61 рисунок, список літератури – 39 найменувань, 7 додатків.

Ключові слова: ФУНКЦІОНАЛЬНИЙ СТАН, КЛАВІАТУРНИЙ ПОЧЕРК, НЕЙРОННА МЕРЕЖА, SELF-ORGANIZING MAP, МЕРЕЖА КОХОНЕНА, VISUAL BASIC, MATLAB, NEURAL NETWORK TRAINING.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ МЕТОДІВ І ЗАСОБІВ ОЦІНКИ ФУНКЦІОНАЛЬНОГО СТАНУ ОПЕРАТОРІВ АВТОМАТИЗОВАНИХ ТЕХНОЛОГІЧНИХ КОМПЛЕКСІВ	9
1.1 Діяльність людини-оператора в автоматизованих технологічних комплексах... 9	
1.2 Задача контролю функціонального стану в системі задач ергономічного проектування автоматизованих технологічних комплексів	11
1.3 Аналіз біометричних методів для оцінки функціонального стану людини оператора.....	11
1.4 Огляд існуючих програмних засобів для оцінки функціонального стану операторів інформаційних систем.....	15
2 ПОСТАНОВКА ЗАДАЧІ ІНФОРМАЦІЙНОЇ СИСТЕМИ	17
2.1 Мета та задачі	17
2.2 Методи дослідження оцінки функціонального стану операторів.....	19
2.3 Вибір засобів реалізації	24
3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОЦІНКИ ФУНКЦІОНАЛЬНОГО СТАНУ ОПЕРАТОРІВ ІНФОРМАЦІЙНИХ СИСТЕМ	26
3.1 Структурно-функціональне моделювання інформаційної технології оцінки функціонального стану операторів	26
3.2 Моделювання варіантів використання.....	34
3.3 Математична модель для оцінки функціонального стану операторів.....	39
4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ОЦІНКИ ФУНКЦІОНАЛЬНОГО СТАНУ ОПЕРАТОРІВ	46
4.1 Результат реалізації інформаційної системи.....	46
4.2 Проведення експериментальних досліджень	51
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	57
ДОДАТОК А. Технічне завдання	62

ДОДАТОК Б. Планування робіт	71
ДОДАТОК В. Лістинг програмного коду	81
ДОДАТОК Г. Інструкція з використання інформаційної системи	104
ДОДАТОК Д. Копія акта впровадження	117
ДОДАТОК Ж. Копії публікацій за темою роботи	118
ДОДАТОК З. Копії дипломів переможця конкурсів студентських наукових робіт	127

ВСТУП

Актуальність теми. Основною вимогою сучасних автоматизованих технологічних комплексів є необхідність виконання ергономічних вимог до робочих місць людей-операторів. Функціональний стан людини суттєво впливає на ефективність діяльності, працеспроможність, безпомилковість, аварійність, фізичний стан. Незважаючи на велику кількість досліджень «людського фактору» задача автоматизованого визначення функціонального стану операторів вирішена не до кінця.

Об'єкт дослідження. Діяльність операторів інформаційних систем.

Предмет дослідження. Автоматизація оцінки функціонального стану операторів.

Мета дослідження. Розробити інформаційну систему оцінки функціонального стану операторів інформаційних систем найбільш оптимальним методом.

У відповідності з метою дослідження ставлять такі **завдання**:

- аналіз методів і засобів визначення психофізіологічного стану операторів;
- формування технічного забезпечення;
- розробка алгоритмів і математичних моделей визначення психофізіологічного стану людини-оператора;
- проектування роботи системи;
- розробка інформаційної системи оцінки функціонального стану людини, що пов'язана з пакетом прикладних програм Matlab;
- проведення експериментального дослідження для формування параметрів оцінки функціонального стану операторів.

Наукова новизна дослідження. На відміну від апаратних методів контролю функціонального стану, запропонований метод використовує модель, основу на використанні апарату нейронних мереж, що забезпечує суттєве зростання точності і характеризується невибагливістю до апаратного забезпечення.

Апробація. Результати дослідження доповідалися на конференціях:

- Міжнародна науково-технічна конференція «Інформатика, Математика, Автоматика ІМА::2018»(Суми, 05-09 лютий 2018 року);
- XII Міжнародна науково-практична конференція «Цифрові технології в освіті, науці, суспільстві» (Петрозаводск, 4-6 грудня 2018 року);
- Міжнародна науково-технічна конференція «Інформатика, Математика, Автоматика ІМА::2019»(Суми, 23-26 лютий 2019 року).

Публікації. За результатами дослідження опубліковано 3 наукові роботи [37, 38, 39]. Копії публікацій наведені в додатку Ж.

Участь в конкурсі студентських наукових робіт. Робота була переможцем:

- II туру Всеукраїнського конкурсу студентських наукових робіт з Комп'ютерної інженерії (Хмельницький, 27-28 березня 2019 року);
- Всеукраїнського конкурсу студентських наукових робіт з напрямку «Інформатика і кібернетика» (Вінниця, 25-26 квітня 2019 року).

Копії дипломів наведені в додатку З.

1 АНАЛІЗ МЕТОДІВ І ЗАСОБІВ ОЦІНКИ ФУНКЦІОНАЛЬНОГО СТАНУ ОПЕРАТОРІВ АВТОМАТИЗОВАНИХ ТЕХНОЛОГІЧНИХ КОМПЛЕКСІВ

1.1 Діяльність людини-оператора в автоматизованих технологічних комплексах

Відмінними ознаками сучасних автоматизованих технологічних комплексів (АТК) [1] є:

- здійснення управління технологічним об'єктом управління (ТОУ) в цілому;
- здійснення управління в темпі протікання технологічного процесу (в реальному масштабі часу ТОУ);
- вибір і реалізація рішень, а також управління ТОУ за участю технічних засобів і людини-оператора.

Об'єктом управління в АТК є окремі енергетичні та технологічні засоби або їх взаємопов'язані комплекси технологічні процеси. Роль ергатичного елемента, який в даному випадку називається відповідно людиною-оператором складається з дистанційного спостереження за роботою і з дистанційного виконання ряду власних функцій (пуск, відключення, перемикання на нові режими, введення керуючих впливів, завантаження деталей в склад і т. д.) Можна виділити наступні типи АТК [2]:

- АТК інформаційного типу;
- АТК комбінованого (інформаційного-який дає поради) типу;
- АТК з прямим цифровим керуванням.

Структура АТК інформаційного типу показана на рис. 1.1(а). а реалізовані нею рівні ієрархії функцій управління - на рис. 1.1(б). Як видно з рис. 1.1(а), оператор

може керувати ТОУ безпосередньо (вручну) і за допомогою системи дистанційного керування (за принципом "включено-виключено") [2, 3].

Переваги цієї структури АТК визначимо за допомогою порівняння її з системами ручного і дистанційного керування технологічними процесами.

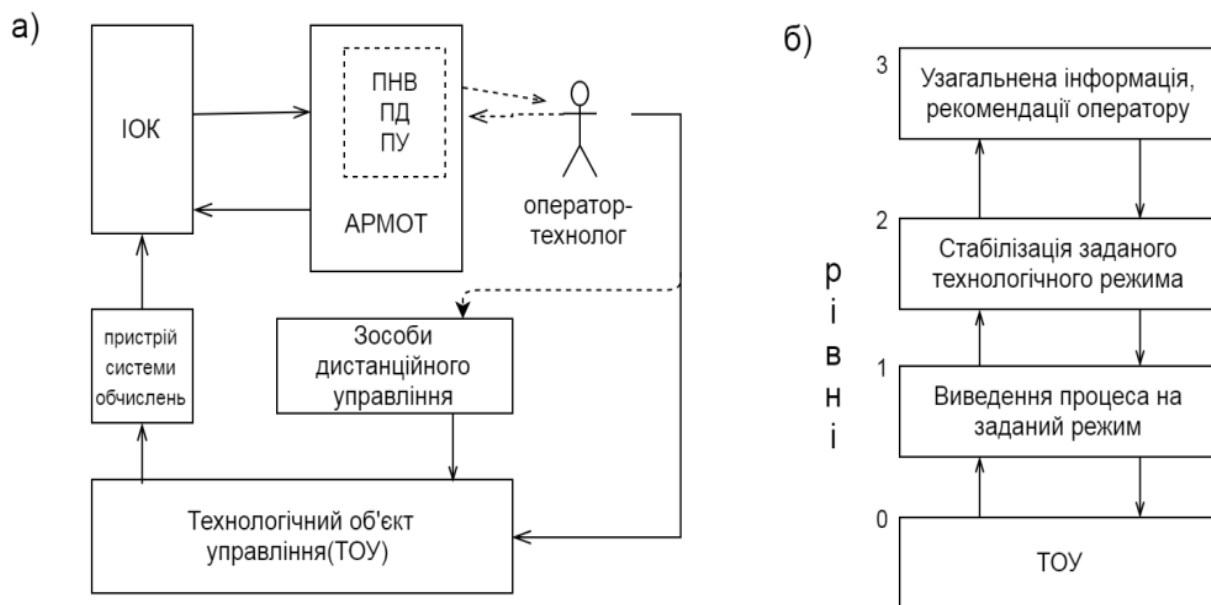


Рисунок 1.1-АТК Інформаційного типу

а) структура АТК; б) рівні ієрархії функцій управління.

До переваг можна віднести можливість: обробки в реальному масштабі часу функціонування ТОУ великого (з урахуванням швидкодії сучасних міні- і мікро-ЕОМ) потоку інформації про хід технологічного процесу і про стан технічних засобів (обладнання) ТОУ; отримання оперативної техніко-економічної інформації про роботу АТК; отримання інформації в концентрованому вигляді на одному пристрої наочного зображення ІОК (у вигляді формуляра, графіка, словесних рекомендацій); документування ходу технологічного процесу і стану обладнання ТОУ в процесі експлуатації; отримання оператором оперативної інформації від ІОК в складних виробничих ситуаціях (в тому числі і аварійних). [2]

1.2 Задача контролю функціонального стану в системі задач ергономічного проектування автоматизованих технологічних комплексів

Технічна частина АТК є лише засобом праці, хоча і кібернетичним, але допоміжним елементом, що сприяє діяльності суб'єкта праці. В цьому суть антропоцентричного підходу до дослідження автоматизованих систем [1].

До складу основних завдань ергономічного проектування входять:

1) вибір чисельності персоналу і розподіл функцій між персоналом системи, тобто розробка загальної організації системи;

2) вибір ступеня автоматизації управління і контролю, тобто розподіл функцій між людиною-оператором і автоматичними пристроями машини на кожному індивідуальному робочому місці;

3) розробка інформаційної моделі і алгоритмів функціонування АТК;

4) проектування індивідуального (колективного) робочого місця;

5) проектування умов праці на робочому місці.

Всі перелічені задачі можуть бути ефективно вирішені лише за умови ефективного контролю за функціональним станом людини-оператора.

Незадовільний функціональний стан може бути причиною помилок, аварій, різноманітних збитків, шкоди здоров'ю.

Таким чином, контроль за функціональним станом є першочерговою задачею оперативного управління АТК.

1.3 Аналіз біометричних методів для оцінки функціонального стану людини оператора

Починаючи з 1960-х років активно розвивається тема комп'ютерної біометрії. Тоді щойно створився біометричний підрозділ Національного інституту стандартів і

технологій США (NIST) [4]. Складність та чималі кошти реалізації технологій не давали людям можливості швидко розповсюджувати нові можливості.

На сьогоднішній день багато де використовуються біометричні технології:

- відсканування пальцю за допомогою додатку в мобільному телефоні;
- віднайти злочинця за допомогою сканеру обличчя;
- ідентифікація людини за голосом або відсканувавши сітчатку ока.

В Європі з'являються біометричні банківські картки з планами розповсюдити їх для створення Національної банківської платформи. За оцінками провідних аналітичних компаній «Tractica» та «J'son & Partners Consulting» здійснено підрахунки щодо темпу росту виручки на ринку біометрії в декілька наступних років. Він становитиме 18,6 % і до 2022 року зросте більше ніж в 2 рази (рис 1.2) [4].

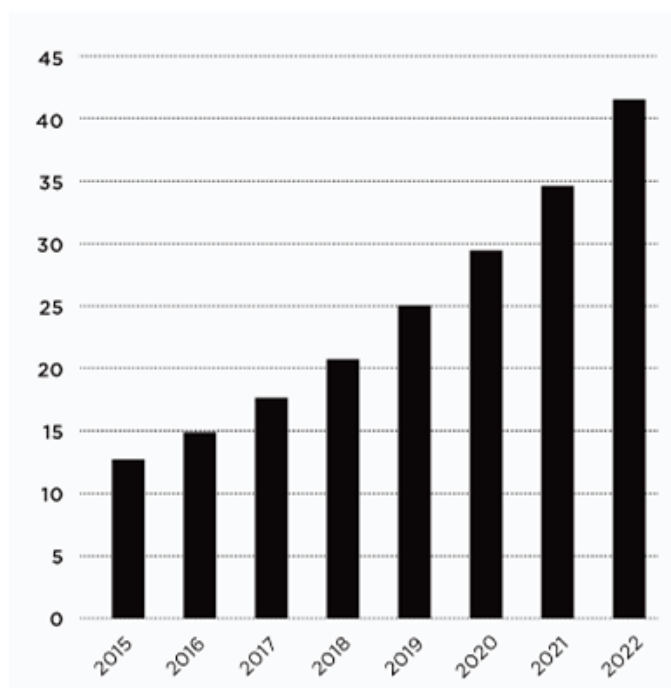


Рисунок 1.2 – Об'єм світового ринку біометричних систем, \$млрд (J'son & Partners Consulting», 2017 р.)

Якщо ж звернути увагу на аутентифікацію, то при розповсюдженні даної біометричної технології, маємо можливість збільшити суму економії, так як зменшиться попит на введення паролів.

Існує ряд біометричних методів аналізу особливостей людини (табл. 1.1):

- ГОЛОС;
- відбиток пальця;
- вени на долоні (а саме їх рисунок);
- радужну оболонку ока;
- обличчя;
- клавіатурний почерк.

Таблиця 1.1 – Біометричні характеристики методів

Метод аналізу особливостей людини	Пояснення	Біометричні технології по типу досліджуваної ознаки
Голос	«-» - можна підробити; - складність в розпізнанні при сторонньому шумі; - зміна голосу (наприклад, хвороба).	статичні
Відбиток пальця	«+» - технологія достатньо безпечна; «-» - папілярний візерунок пальця нестабільний.	статичні
Рисунок вен на долоні	«+» - складність втратити або підробити; «-» - дороге використання; - при захворюваннях крові або фізичних навантаженнях можуть виникати проблеми з розпізнанням.	статичні
Радужна оболонка ока	«+» - висока точність результату при використанні обох очей одразу; «-» - занадто дорого.	статичні

Продовження таблиці 1.1.

Метод аналізу особливостей людини	Пояснення	Біометричні технології по типу досліджуваної ознаки
3D – зображення обличчя	«+» - у відкритому доступі; - без затрат грошей на додаткове обладнання; «-» - ризик компрометації.	статичні
Клавіатурний почерк	«+» - дешево; - без затрат грошей на додаткове обладнання; «-» - постійна зміна набору	динамічні

Перелічені вище технології використовують різну реалізацію за допомогою математичних моделей. Розробники використовують крім різновид них алгоритмів ще й можливість захисту від заміни, шахрайства.

Для вирішення оптимальної технології для оцінки функціонального стану людини за біометричними показниками проведено ряд тестувань (рис. 1.3). Велику роль грає швидкість результату, але в даному випадку вона не враховувалася.

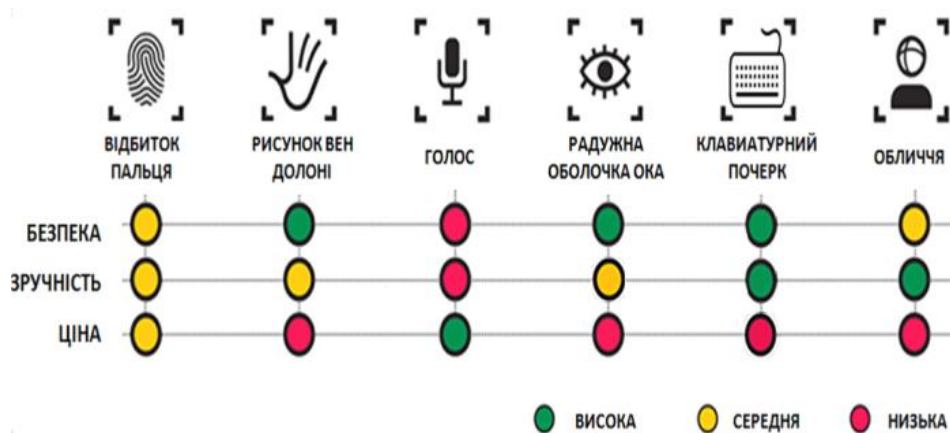


Рисунок 1.3 – Порівняльна оцінка технологій біометричної аутентифікації

Значну перевагу серед методів біометричних технологій отримують динамічні методи, адже вони дозволяють змінювати біометричний еталон (створені персональні дані конкретної людини) не витрачаючи на це великі кошти.

Серед динамічних методів, які побудовані на психофізіологічній особливості людини, вирізняють такі види розпізнавання:

- рукописний почерк;
- клавіатурний почерк;
- голосові характеристики.

Сучасні дослідження виділяють розпізнавання клавіатурного почерку, як біометричний метод, що має стабільність [8] порівняно з іншими двома. Методи розпізнавання рукопису або голосу мають ряд недоліків, наприклад:

- потребують великих затрат на реалізацію;
- рукопис швидко змінюється, тому потребуватиме постійних тренувань на пристрої;
- швидкий обхід перевірки функціонального стану методом голосових характеристик за допомогою запису звука.

У результаті проведеного аналізу видно, що оптимальним біометричним методом для оцінки функціонального стану оператора інформаційних систем є саме метод аналізу клавіатурного почерку.

1.4 Огляд існуючих програмних засобів для оцінки функціонального стану операторів інформаційних систем

Є безліч досліджень та намагань реалізувати програмний продукт, який дасть можливість оцінити функціональний стан операторів інформаційних систем, але ці спроби не дійшли в експлуатацію.

Для аналізу обрано підпрограмну реалізацію студентки Кривомаз Д.О. Сумського державного університету кафедри комп'ютерних наук. Ця підсистема

розроблялася для системи електронного навчання в 2015 році, але так і не увійшла до неї. Одним із головних причин відмови від цього задуму є неможливість пристосування підсистеми до кожного з користувачів.

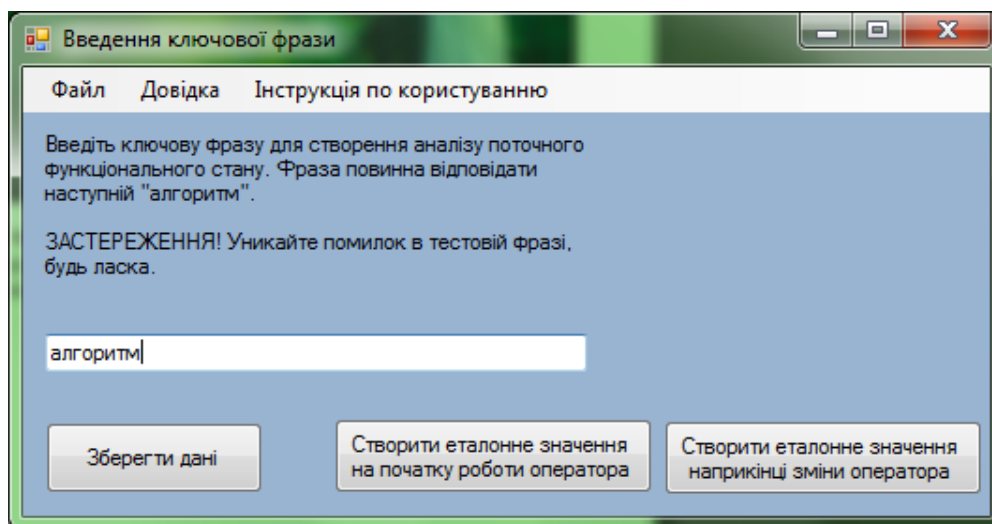


Рисунок 1.4 – Головне вікно підсистеми «klaviatura project»

Дослідивши підпрограму зсередини та зовні (рис. 1.4) зроблено висновки, які вказують як на переваги, так і на слабкі сторони підсистеми.

Підсистема має дві *сильні* сторони: по-перше, працює самостійно, не потребує інших програмних продуктів; по-друге, простий у використанні інтерфейс.

Слабкі сторони підпрограми: неможливість використання підсистеми декількома операторами на одному комп'ютері або ноутбуці; створення еталонного значення на початку або наприкінці робочої зміни є даремним, адже вхідні дані від контрольного слова записуються до окремого файлу, який ніде не використовується; недалекоглядний проект, адже перевірка функціонального стану йде за контрольним словом, яке не змінюється, що дає змогу користувачу вивчити його введення; використання нейронної мережі з вчителем, яка сформована та навчена ще до створення еталонного значення; немає автоматизації на ведення статистичних даних, що містять результати кожної пройденої перевірки.

2 ПОСТАНОВКА ЗАДАЧІ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Мета та задачі

Інформаційна система повинна забезпечувати автоматизацію оцінки функціонального стану операторів інформаційних систем, а саме збереження введених даних у файли, створення та навчання нейронної мережі за допомогою цих даних, перевірку функціонального стану за порівнянням навченої мережі та нововведеними значеннями, вивід і збереження результатів до файлу зі статистичними даними.

Система створена для підвищення ефективності діяльності людини при різних фізичних відхиленнях – загалом, перевтоми.

Основна задача: розробити технологію з використанням нейронної мережі (так як вона є більш стійкою та точною) для оцінки функціонального стану людини-оператора. Детальна інформація щодо задач даного проекту розміщена в додатку А.

Нейронна мережа виконує задачу кластеризації та ділить функціональний стан операторів на 2 кластери («задовільний» та «незадовільний»). Для даної задачі обрано нейронну мережу Кохонена, а точніше самоорганізовану мапу Кохонена, яка проводить навчання без вчителя та використовує вхідні дані, які користувач має вводити з клавіатури.

Реалізовано введення 10 слів із 10 літер, які розміщені в хаотичному порядку на клавіатурі, та викликаються по черзі при відкритті вікна перевірки поточного стану.

Вхідними даними системи для перевірки функціонального стану виступають дані слів, які зчитано з клавіатури, – час натискання на клавішу та час утримання цієї ж клавіші. Загалом за одне введення контрольного слова з клавіатури до нейронної мережі потрапляє 20 значень (по 10 даних для натискання та утримання) (рис. 2.1).

Вихідними даними системи є значення, яке знайшлося після симуляції нейронної мережі. Якщо результат перевірки показує більшу приналежність вхідних значень до першого кластеру (сегменту) зі значенням 0, то система видасть повідомлення про задовільний стан оператора. Якщо ж значення 1 – другий кластер (сегмент), повідомлення про незадовільний стан користувача.

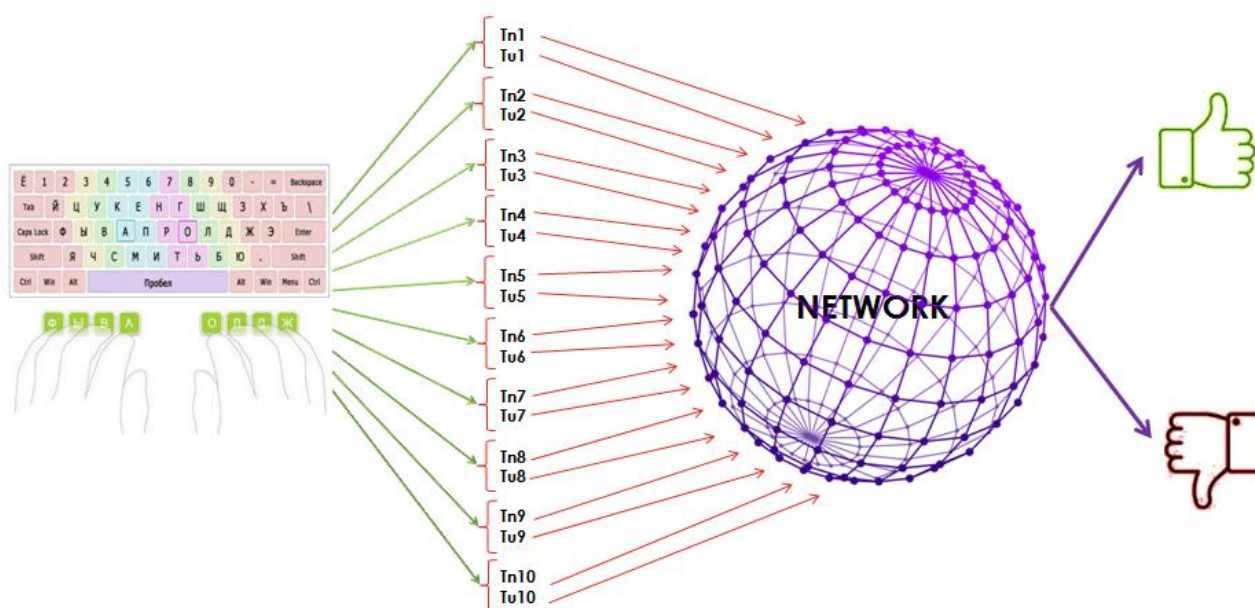


Рисунок 2.1 – Модель роботи системи

(tn_i – час натискання на клавішу, tu_i – час утримання клавіші)

Результати розробки інформаційної системи мають ряд необхідних функцій для оцінки функціонального стану операторів інформаційних систем:

- введення контрольних слів для навчання нейронної мережі та симуляції з нею;
- збереження зчитаних даних із клавіатури до текстових файлів;
- створення та видалення директорії, де розміщуються результати роботи програми;
- створення та навчання само організованої мапи Кохонена;
- виведення результатів симуляції у вигляді інформаційного повідомлення та їх запис до текстового файлу;

- створення, додавання та видалення результатів системи про оцінку функціонального стану людини-оператора;
- ведення статистичних даних оснований на поточних перевірках функціонального стану користувача.

2.2 Методи дослідження оцінки функціонального стану операторів

КЛАВІАТУРНИЙ ПОЧЕРК Є ОДНИМ З ПОПУЛЯРНИХ МЕТОДІВ ІДЕНТИФІКАЦІЇ ЛЮДЕЙ-ОПЕРАТОРІВ, ОСКІЛЬКИ НЕ ІСНУЄ КОРИСТУВАЧІВ З ІДЕНТИЧНИМ КОМП'ЮТЕРНИМ ПОЧЕРКОМ.

Клавіатурний почерк – це набір динамічних характеристик роботи на клавіатурі [9].

Основними відмінними характеристиками клавіатурного почерку є час утримання клавіші і паузи між натисканнями.

Час утримання клавіші - це період, в протязом якого клавіша знаходиться в натиснутому стані. Вимірюється, як правило, в мілісекундах.

Накладення натискань клавіш відбувається тоді, коли одна клавіша ще не відпущена, а інша вже натискається. З підвищенням швидкості набору тексту збільшується число накладень.

Пауза між натисканням - це тимчасової інтервал, коли одна клавішу вже відпущена, а наступна ще не натиснута.

Для ідентифікації клавіатурного почерку має сенс порівнювати динамічні характеристики натискань тільки на клавіші з буквами, цифрами та знаками пунктуації, оскільки натискання на системні клавіші, наприклад, Alt або Esc, як правило, не належать до набору тексту.

Алгоритми розпізнавання клавіатурного почерку можна розділити на 3 групи:

- алгоритми, які аналізують почерк під час введення контрольного слова;
- алгоритми, які аналізують почерк після введення текстового фрагмента або фрази;

- алгоритми, які постійно проводять прихований моніторинг клавіатурного почерку користувача.

Алгоритми першої групи забезпечують найбільшу швидкість: користувачеві потрібно тільки ввести слово. Однак точність в цьому випадку невисока, особливо в разі невеликої кількості літер.

Алгоритми другої групи дозволяють забезпечити більшу точність, в порівнянні з першою групою. Однак на введення довгого слова потрібен час, що може викликати негативні емоції у користувача, особливо в разі, якщо йому часто доводиться проходити тестування поточного функціонального стану.

Алгоритми третьої групи можуть забезпечити високу точність. При цьому вони вимагають більше ресурсів.

У роботі використана друга група для створення навчальної вибірки. Порівняння характеристик клавіатурного почерку більш вдало відбувається за рахунок використання вірогідно-статистичних методів і за допомогою нейронних мереж.

Одним із методів покращення роботи алгоритму є створення та постійне редагування еталонних значень. Ця дія дозволить вхідним даним не застаріти та завжди відповідати поточному рівню швидкості друку людини-оператора.

Вважається, що методи, засновані на використанні нейронних мереж, забезпечують більш високу точність, але потребують великих обчислювальних потужностей.

Класичне навчання нейронних мереж ділиться на дві категорії (рис. 1.6):

- з вчителем (Supervised Learning);
- без вчителя (Unsupervised Learning).

Навчання з вчителем розглядає задачі з вже вказаним методом їх рішення. Наприклад, розподіл картинок різних звірів відбувається за рахунок вже існуючих критеріїв оцінки кого куди віднести, тобто існуючих класів тварин.

Навчання без вчителя – самоорганізоване. Наприклад, машина отримує завдання розподілити ті ж самі картинки, але порівняно з вище вказаним методом,

вона сама розрізняє закономірності рисунків та розподіляє їх на ті ж самі групи без ніяких попередньо заданих критеріїв.



Рисунок 2.2 – Категорії класичного навчання

Навчання з вчителем є більш точним та швидшим, якщо розглядати бойові задачі. Інший вид – навчання без вчителя – з точки зору біологічних коренів нейронних мереж є більш правдоподібним для моделі навчання [10].

Neural Network Toolbox включає два типи самоорганізованих неконтрольованих мереж з навчанням без вчителя:

- конкурентні шари;
- самоорганізовані карти (мапи) [11].

Ці два типи входять до нейронної мережі Кохонена (рис. 2.3). Різниця між ними лише в тому, що конкурентні шари Кохонена – мережа з нейронами, які неупорядковано, а самоорганізована мапа Кохонена – мережа з нейронами, які впорядковано.

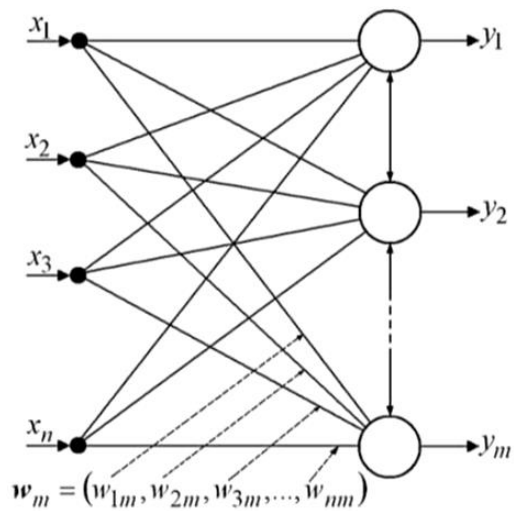


Рисунок 2.3 – Структура мережі Кохонена

Для формування мережі Кохонена необхідно:

- вектор вхідних сигналів:

$$x(i) = (x_1(i), x_2(i), \dots, x_n(i)), \quad (2.1)$$

де $i = 1, 2, \dots, P$;

- вихідний вектор:

$$y = (y_1, y_2, \dots, y_m), \quad (2.2)$$

де m – кількість кластерів;

- матриця множини вагових коефіцієнтів:

$$W = \begin{pmatrix} W^{11} & W^{12} & \dots & W^{1m} \\ W^{21} & W^{22} & \dots & W^{2m} \\ \vdots & \vdots & \dots & \vdots \\ W^{k1} & W^{k2} & \dots & W^{km} \end{pmatrix}, \quad (2.3)$$

де $W^{ij} = (w_1^{ij}, w_2^{ij}, \dots, w_n^{ij})$ – елементи матриці (вектори вагових коефіцієнтів).

Для вибору типу мережі для системи оцінки функціонального стану операторів інформаційних систем було визначено помилки функціонування кожної з них. Для цього зроблено декілька кроків.

1. Задано однакову навчальну вибірку, яка складалася з значень натиснутих клавіш (часу натискання та утримання).
2. Створено нейронну мережу Кохонена, окремо для кожного з типів.
3. Навчено її за допомогою функції `train` (так як має низку переваг порівняно з `adapt`).
4. Визначено похибку.

Порівнюючи результати отриманих помилок отримано, що для визначення функціонального стану людини слід обрати Самоорганізовану мапу Кохонена, адже, порівнюючи її з іншим типом, помилка була нижчою (табл. 2.1).

Таблиця 2.1 – Порівняльна таблиця нейронних мереж

Тип мережі	Функція в Matlab	Помилка
Конкурентний шар Кохонена	<code>newc</code>	[3.104e-05]
Самоорганізована мапа Кохонена	<code>newsom</code>	[2.8276e-09]

Різновид мережі Кохонена – само організована мапа Кохонена (у Neural Network Toolbox – Self-Organizing Map або SOM) – проводить кластеризацію об'єктів і виконує візуалізацію результатів багатовимірним проектуванням.

Дану мережу застосовують не тільки для візуалізації результатів аналізу даних та кластеризації, а й для побудови структури багатовимірних даних [13].

Self-Organizing Map схожий своїм алгоритмом з k-means (алгоритм k – найближчих середніх). Основною ознакою SOM є те, що всі вузли (нейрони) впорядковані в структуру у вигляді двовимірної сітки, в якій при навчанні нейронної мережі відбувається модифікація нейрона-переможця та його сусідів. Як результат даний тип мережі вважається одним із методів проектування багатовимірної площі в площу з більш меншою розмірністю [14].

2.3 Вибір засобів реалізації

Для реалізації програмного продукту використано мову програмування Microsoft Visual Basic 2010, пакет прикладних програм Matlab R2011b, табличний процесор Microsoft Excel 2010 та текстові файли.

Microsoft Visual Basic це відмінна платформа для написання різних видів додатків, які мають доступ до бази даних. Це власна мова програмування Microsoft.

Дану мову програмування обрано за її сильні сторони:

- Visual Basic - додатки керуються подіями;
- Visual Basic підтримує принципи об'єктно-орієтованого дизайну;
- Visual Basic – повна система розробки додатків для Windows;
- Visual Basic використовує елементи керування ActiveX, динамічно пов'язані бібліотеки (DLL) та надбудови.

Також Visual Basic для додатків є версією, яка використовується для програмування програм Office, наприклад Access, Excel, Word та PowerPoint.

Matlab – програмне забезпечення, яке може бути використане в задачах багатоцільового призначення.

Наприклад, до його можливостей входить обробка зображень, автоматизація, обробка сигналів і багато інших функцій.

Він має ряд переваг серед інших методів і мов. Його основним елементом даних є матриця.

Багатофункціональність Matlab дозволяє створювати окремий програмний продукт, користуватися штучним інтелектом за допомогою **Neural Network Toolbox**, проводити дослідження як математичних функцій так і вбудованих.

Neural Network Toolbox функція виклику в Matlab – nntool в Command Window – відкриває вікно Мережі (NETWORK) або Менеджер даних, у якому дозволено імпортувати, створювати, використовувати та експортувати нейронні мережі та дані, що з ними пов'язані.

Дипломний продукт використовує функції нейронних мереж для виконання задачі оцінки функціонального стану операторів інформаційних систем.

Пакет Matlab містить широкий спектр активаційних та навчальних функцій, також він може бути пов'язаний з іншими офісними програмами, наприклад MS Word, MS Excel, текстовими редакторами та іншими прикладними програмами.

Microsoft Excel - програма для створення і роботи з електронними таблицями, за допомогою якої зручно робити обчислення різної складності.

Таблиці спрощують сортування, фільтрацію і форматування даних на аркуші. Зведені таблиці спрощують уявлення, зведення та деталізацію складних даних.

Текстовий файл є своєрідним комп'ютерним файлом, який структурований як послідовність рядків електронного тексту. У даній роботі використовується як місце обміну даними між програмами Microsoft Visual Basic та Matlab.

3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОЦІНКИ ФУНКЦІОНАЛЬНОГО СТАНУ ОПЕРАТОРІВ ІНФОРМАЦІЙНИХ СИСТЕМ

3.1 Структурно-функціональне моделювання інформаційної технології оцінки функціонального стану операторів

Для структурно-функціонального моделювання використано технологію структурного аналізу та проектування (SADT). Методи SADT базуються на декількох концепціях. Декомпозиція зверху вниз дозволяє обробляти інформацію за прогресивними рівнями деталізації. Формування даної моделі сприяє кращому розумінню роботи та мети проекту в цілому. Прийняття різноманітних додаткових точок зору дозволяє враховувати всі відповідні аспекти системи, обмежуючи при цьому розгляд у будь-який час до однієї чітко визначеної теми. Процедури розгляду та ітерацій забезпечують якість розробленої моделі [21].

Графічна мова SADT складається з блоків та стрілок, які використовуються при побудові діаграм. Дана мова стосується тільки структурованої декомпозиції предмету. Схема може моделювати процеси або дані.

Діаграма, яка моделює процеси, називається Actigram, використовує блоки для показу окремих процесів і використовує стрілки для зображення потоків даних між процесами, будь-яких обмежень, що застосовані, і механізми для виконання процесів.

Стрілки, що входять і виходять з блоків, служать для прив'язки контексту процесу, і це може бути розкладено на подальших Actigram через послідовні рівні деталізації до будь-якого необхідного рівня.

Для створення структурно-функціональної моделі використано програмний продукт Erwin Process Modeler.

Для створення контекстної діаграми та подальшої деталізації використано методологію IDEF0. Даний підхід заснований на загальному описі та моделюванні

функції методології бізнес процесів. За допомогою методології створено функціональні моделі, які представляють структуру та функцію системи [22].

Розглянемо процес роботи інформаційної технології «Визначення ФС». На рис. 3.1 представлено контекстну діаграму процесу.

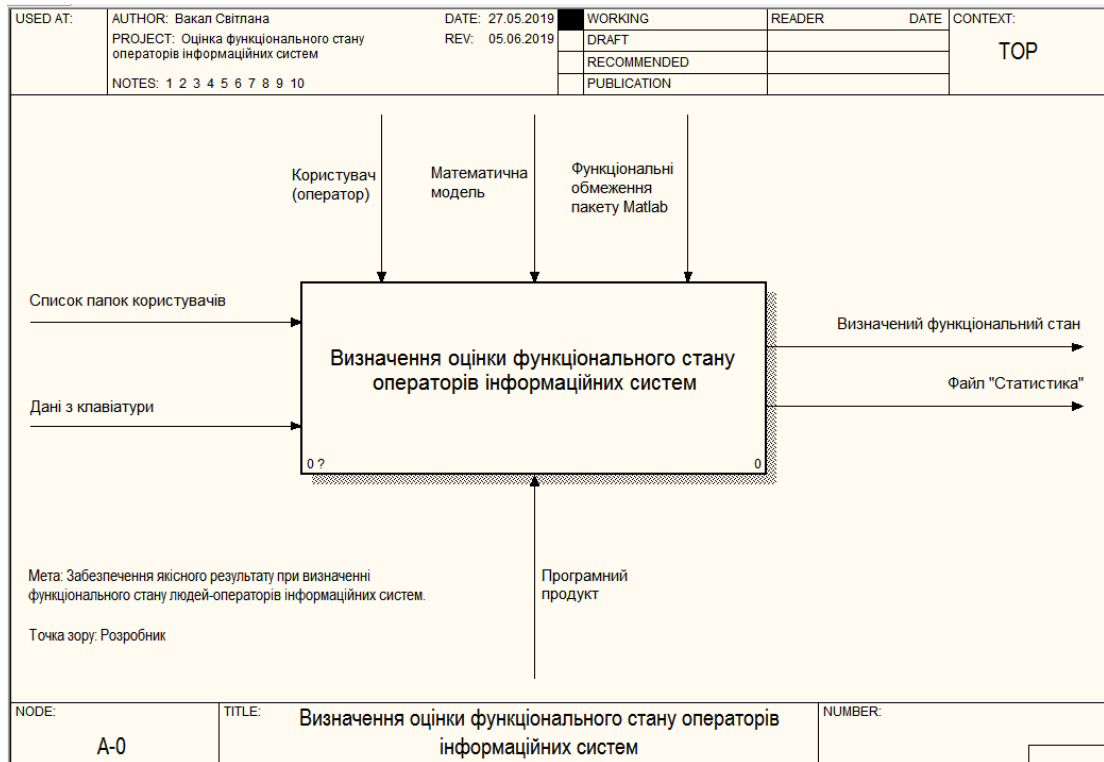


Рисунок 3.1 – Контекстна діаграма

Вхідні дані до функції «Визначення оцінки функціонального стану операторів інформаційних систем»:

- Список папок користувачі (вибір папки зі списку раніше створених);
- Дані з клавіатури (зчитування системою час натискання та час утримання клавіші введеного слова).

Елементи керування функції «Визначення оцінки функціонального стану операторів інформаційних систем»:

- Користувач (оператор) (людина, що працюватиме з програмним продуктом);
- Математична модель;
- Функціональні обмеження пакету Matlab.

Механізмом функції «Визначення оцінки функціонального стану операторів інформаційних систем» є сам програмний продукт, адже всі дії відбуваються в його рамках.

Вихідними значеннями функції «Визначення оцінки функціонального стану операторів інформаційних систем»:

- Визначений функціональний стан («задовільний» або «незадовільний»);
- Файл «Статистика» (файл у якому зберігаються всі перевірки поточного функціонального стану оператора).

Та як контекстна діаграма – це узагальнений опис інформаційної системи, то як наслідок слід розробити декомпозицію для деталізації структури.

Результатом декомпозиції стало створення трьох блоків у діаграмі декомпозиції нульового рівня (рис. 3.2).

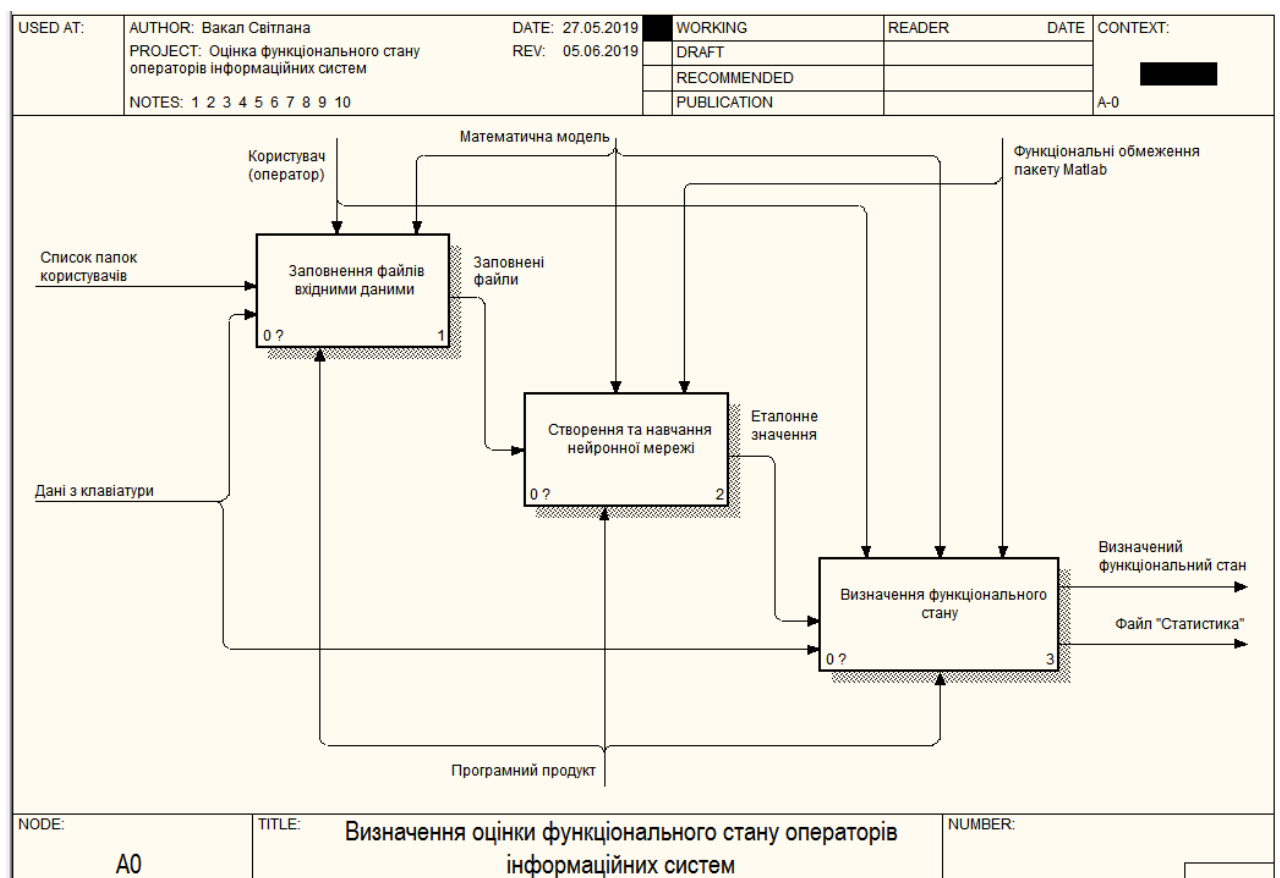


Рисунок 3.2 – Діаграма декомпозиції IDEF0

Діаграма декомпозиції A0 методології IDEF0 складається з блоків:

- Заповнення файлів вхідними даними;
- Створення та навчання нейронної мережі;
- Визначення функціонального стану.

Таблиця 3.1 – Пояснення діаграми декомпозиції нульового рівня

Діяльність	Класи стрілок	Стрілки, що входять до класів
«Заповнення файлів вхідними даними»	Стрілки входу	1. Список папок користувачів 2. Дані з клавіатури
	Стрілки керування	1. Користувач 2. Математична модель
	Стрілки механізму	Програмний продукт
	Стрілки виходу	Заповнені файли
«Створення та навчання нейронної мережі»	Стрілки входу	Заповнені файли
	Стрілки керування	1. Математична модель 2. Функціональні обмеження пакету Matlab
	Стрілки механізму	Програмний продукт
	Стрілки виходу	Еталонне значення
«Визначення функціонального стану»	Стрілки входу	1. Еталонне значення 2. Дані з клавіатури
	Стрілки керування	1. Користувач (оператор) 2. Математична модель 3. Функціональні обмеження пакету Matlab
	Стрілки механізму	Програмний продукт
	Стрілки виходу	1. Визначений функціональний стан 2. Файл «Статистика»

Контекстна діаграма має три рівні декомпозиції (рис. 3.3 – 3.5).

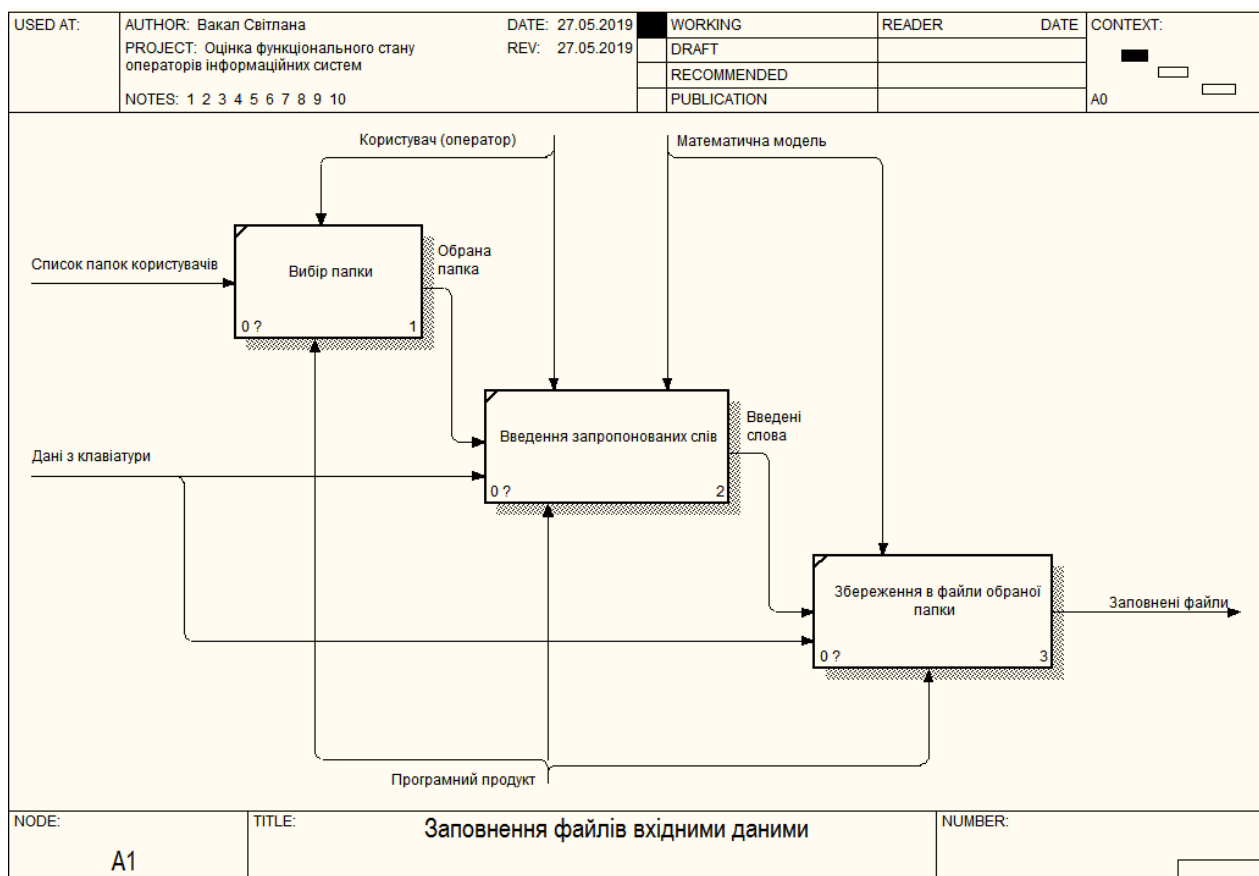


Рисунок 3.3 – Діаграма декомпозиції IDEF0 A1 «Заповнення файлів вхідними даними»

Діаграма декомпозиції A1 методології IDEF0 складається з блоків:

- Вибір папки;
- Введення запропонованих слів;
- Збереження в файл обраної папки.

Таблиця 3.2 – Пояснення діаграми декомпозиції першого рівня

Діяльність	Класи стрілок	Стрілки, що входять до класів
«Вибір папки»	Стрілки входу	Список папок користувачів
	Стрілки керування	Користувач
	Стрілки механізму	Програмний продукт
	Стрілки виходу	Обрана папка

Продовження таблиці 3.2.

Діяльність	Класи стрілок	Стрілки, що входять до класів
«Введення запропонованих слів»	Стрілки входу	1. Обрана папка 2. Дані з клавіатури
	Стрілки керування	1. Користувач 2. Математична модель
	Стрілки механізму	Програмний продукт
	Стрілки виходу	Введені слова
«Збереження в файл обраної папки»	Стрілки входу	1. Введені слова 2. Дані з клавіатури
	Стрілки керування	Математична модель
	Стрілки механізму	Програмний продукт
	Стрілки виходу	Заповнені файли

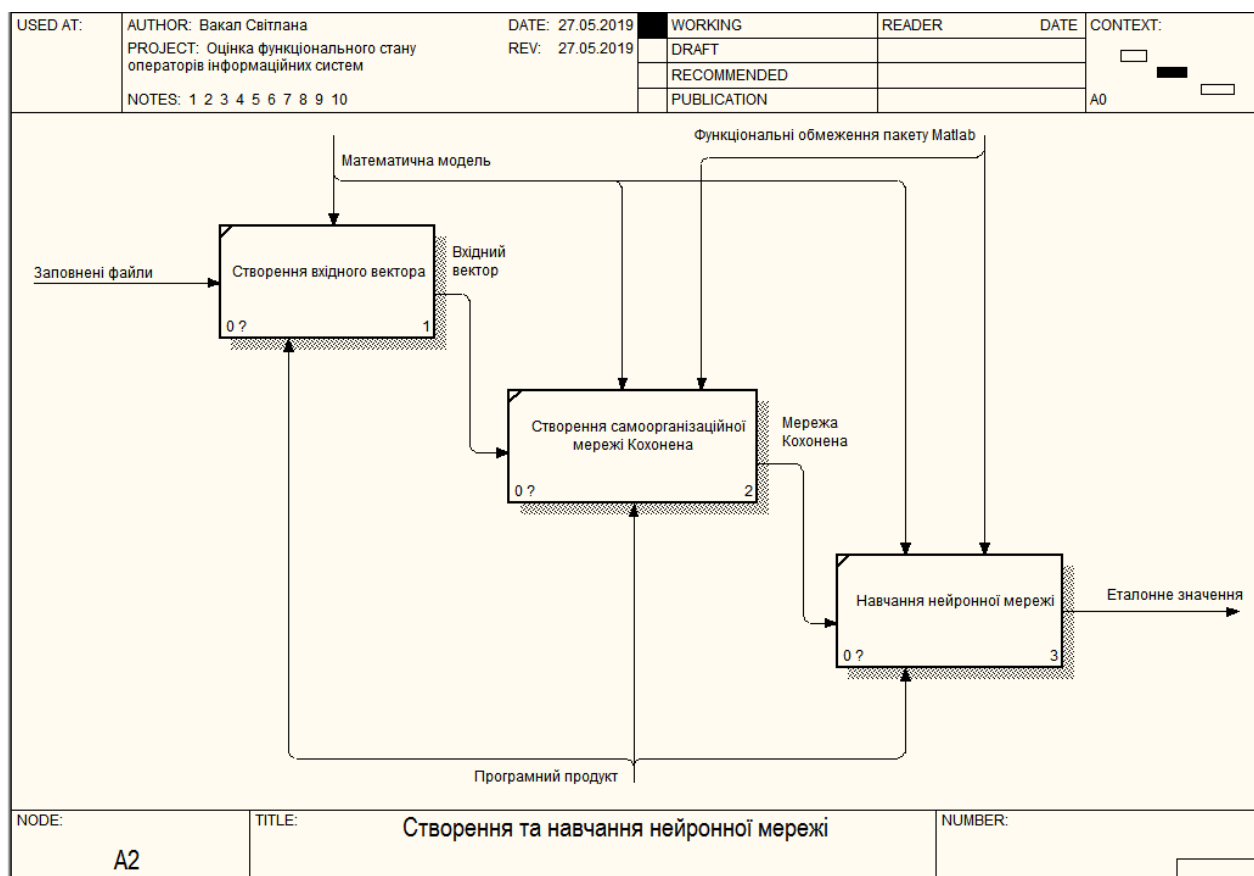


Рисунок 3.4 – Діаграма декомпозиції IDEF0 A2 «Створення та навчання нейронної мережі»

Діаграма декомпозиції A2 методології IDEF0 складається з блоків:

- Створення вхідного вектора;
- Створення самоорганізаційної мережі Кохонена;
- Навчання нейронної мережі.

Таблиця 3.3 – Пояснення діаграми декомпозиції другого рівня

Діяльність	Класи стрілок	Стрілки, що входять до класів
«Створення вхідного вектора»	Стрілки входу	Заповнені файли
	Стрілки керування	Математична модель
	Стрілки механізму	Програмний продукт
	Стрілки виходу	Сформований вхідний вектор
«Створення самоорганізаційної мережі Кохонена»	Стрілки входу	Сформований вхідний вектор
	Стрілки керування	1. Математична модель 2. Функціональні обмеження пакету Matlab
	Стрілки механізму	Програмний продукт
	Стрілки виходу	Мережа Кохонена
«Навчання нейронної мережі»	Стрілки входу	Мережа Кохонена
	Стрілки керування	1. Математична модель 2. Функціональні обмеження пакету Matlab
	Стрілки механізму	Програмний продукт
	Стрілки виходу	Еталонне значення

Діаграма декомпозиції A3 методології IDEF0 складається з блоків:

- Введення слова для перевірки з клавіатури;
- Порівняння значень;
- Запис до файлу результату;
- Виведення повідомлення про поточний функціональний стан;

- Запис до Excel-файлу результатів перевірки.

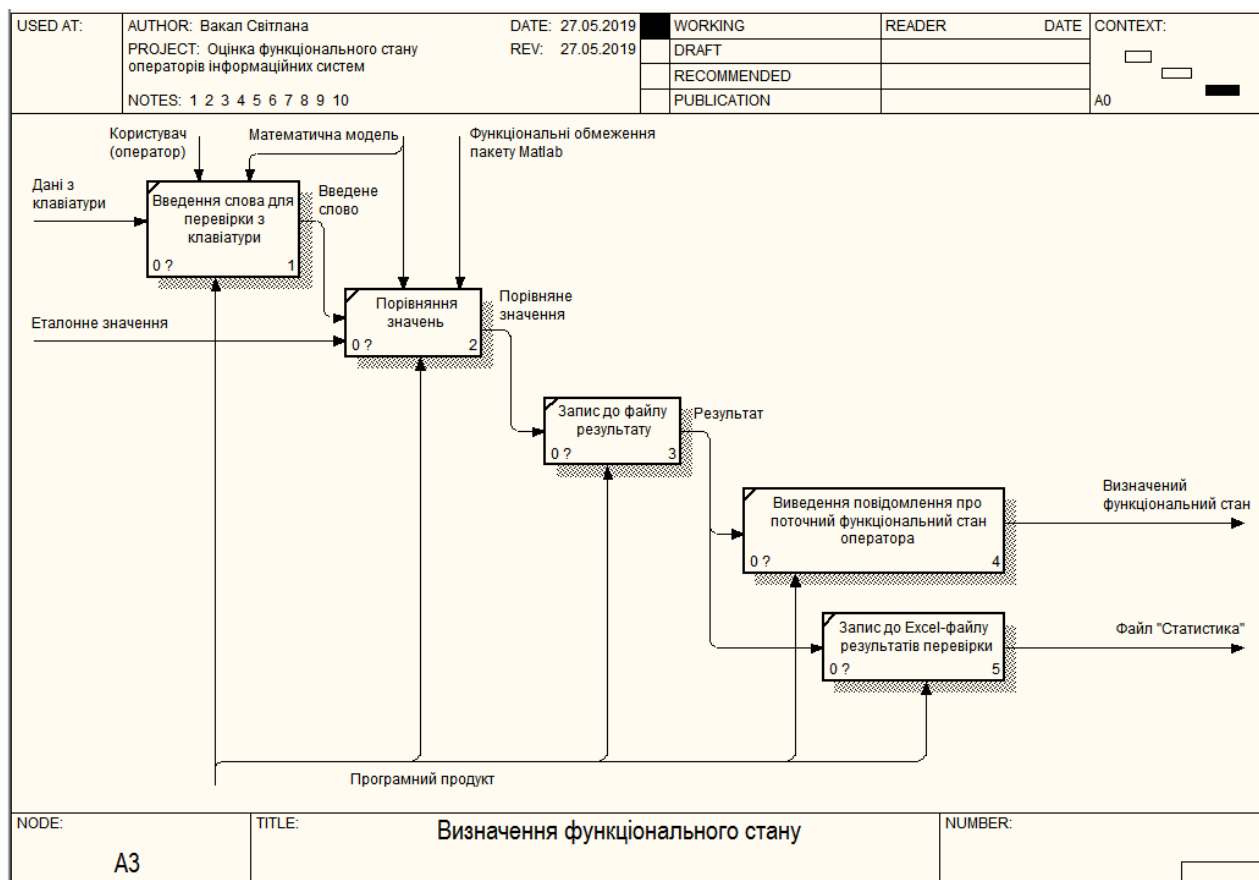


Рисунок 3.5 – Діаграма декомпозиції А3 «Визначення функціонального стану»

Таблиця 3.4 – Пояснення діаграми декомпозиції третього рівня

Діяльність	Класи стрілок	Стрілки, що входять до класів
«Введення слова для перевірки з клавіатури»	Стрілки входу	Дані з клавіатури
	Стрілки керування	1. Користувач (оператор) 2. Математична модель
	Стрілки механізму	Програмний продукт
	Стрілки виходу	Введене слово
«Порівняння значень»	Стрілки входу	1. Введене слово 2. Еталонне значення
	Стрілки керування	1. Математична модель 2. Функціональні обмеження пакету Matlab

Продовження таблиці 3.4.

Діяльність	Класи стрілок	Стрілки, що входять до класів
	Стрілки механізму	Програмний продукт
	Стрілки виходу	Порівняне значення
«Запис до файлу результату»	Стрілки входу	Порівняне значення
	Стрілки керування	---
	Стрілки механізму	Програмний продукт
	Стрілки виходу	Результат
«»	Стрілки входу	Результат
	Стрілки керування	---
	Стрілки механізму	Програмний продукт
	Стрілки виходу	Визначений функціональний стан
«»	Стрілки входу	Результат
	Стрілки керування	---
	Стрілки механізму	Програмний продукт
	Стрілки виходу	Файл «Статистика»

3.2 Моделювання варіантів використання

Варіант використання, прецедент (Use Case) – це програмний і системний термін, який описує, як користувач використовує систему для досягнення певної мети. Use Case виступає в якості методики моделювання програмного забезпечення, що визначає функції, які необхідно реалізувати, і вирішення будь-яких помилок, які можуть виникнути [23].

Характеристики, які пов'язані з Use Case:

- організація функціональних вимог;
- моделювання цілей взаємодії системи користувача;
- запис сценаріїв від тригерів до кінцевих цілей;

- опис основного курсу дій і виняткового потоку подій;
- користувач має доступ до функціональності іншої події [23].

Актори:

1. Користувач – людина-оператор, який працює з даним програмним продуктом.
2. Текстовий файл – файл, у якому міститься інформація про введені з клавіатури слова, а саме час натискання та час утримання на клавішу в мілісекундах.
3. MS Excel – програма Microsoft Excel 2010 – файл із статистичними даними про користувача, який перевіряє поточний функціональний стан.

Варіанти використання:

1 Потік подій ВВ Головна сторінка (1 Головна сторінка)

Передумови: програма запущена, форма «Головна сторінка» відкрита.

Процес: користувач має змогу переглянути список існуючих папок, які створили інші користувачі або він сам раніше. Натиснувши на одну з назв папки користувач потрапляє до проміжного вікна з вибором «Навчити під себе НМ» або «Перевірити ФС».

Післяумови: форма проміжного вікна з вибором клавіш «Навчити під себе НМ» або «Перевірити ФС» відкрита.

2 Потік подій ВВ Додати папку (2 Додати папку)

Передумови: форма «Головна сторінка» відкрита, натиснута позначка додавання папки.

Процес: користувач вводить назву нової папки натискає кнопку «Додати папку», програма записує новостворену папку до випадних списків програми та у директорію за адресою «C:\FS».

Післяумови: список даних про папки оновлено та новоутворена папка збережена до директорії за адресою «C:\FS».

3 Потік подій ВВ Видалити папку (3 Видалити папку)

Передумови: форма «Головна сторінка» відкрита, натиснута позначка видалення папки.

Процес: користувач обирає з випадного списку папку, яку бажає видалити, натискає кнопку «Видалити папку», програма видаляє обрану папку з випадних списків програми та з директорії за адресою «C:\FS».

Післяумови: список даних про папки оновлено та папку видалено з директорії за адресою «C:\FS».

4 Потік подій ВВ Про автора (4 Про автора)

Передумови: форма «Головна сторінка» відкрита.

Процес: користувач натискає на кнопку «Про автора» і відкривається форма «Про автора».

Післяумови: форма «Про автора» відкрита.

5 Потік подій ВВ (5 Обрати дію)

Передумови: ВВ Головна сторінка (1 Головна сторінка) виконаний.

Процес: користувач обирає, яку подію йому слід виконати «Навчити під себе НМ» або «Перевірити ФС».

Післяумови: обрано одну з запропонованих подій «Навчити під себе НМ» або «Перевірити ФС».

6 Потік подій ВВ Персональне налаштування системи(6 Навчання НМ)

Передумови: ВВ Головна сторінка(1 Головна сторінка) виконаний, натиснута кнопка «Навчити під себе НМ», ВВ (5 Обрати подію) виконаний.

Процес: користувач вводить запропоновані слова та натискає на кнопку запису в файл, зчитані з клавіатури дані записуються окремо, у відповідності зі словом, до файлів.

Післяумови: дані зчитані з клавіатури записано до файлів.

7 Потік подій ВВ Перевірка поточного ФС (7 Перевірка ФС)

Передумови: ВВ Головна сторінка(1 Головна сторінка) виконаний, натиснута кнопка «Перевірити ФС», ВВ (5 Обрати подію) виконаний.

Процес: відкриття вікна «Перевірка поточного ФС».

Післяумови: вікно «Перевірка поточного ФС» відкрито.

8 Потік подій ВВ Зберегти в файл(8 Зберегти в файл)

Передумови: форма «Персональне налаштування системи» відкрита.

Процес: користувач вводить перелічені слова в текстових полях та натискає кнопки додавання в файл, зчитані з клавіатури дані зберігаються до файлів.

Післяумови: збережено файли зі зчитаними даними про час натискання та утримання клавіші.

9 Потік подій ВВ Видалити файл (9 Видалити файл)

Передумови: ВВ Персональне налаштування системи(6 Навчання НМ) виконаний.

Процес: користувач натискає кнопку з позначкою видалення файлів, після натискання перехід до вікна «Видалення файлів із вхідними даними».

Післяумови: вікно «Видалення файлів із вхідними даними» відкрито.

10 Потік подій ВВ Видалення файлів із вхідними даними (10 Видалити 1 файл)

Передумови: ВВ Видалити файл (9 Видалити файл) виконаний.

Процес: користувач обирає з випадного списку файл, який необхідно видалити та натискає на кнопку видалення, файл видаляється з директорії та оновлюється випадний список.

Післяумови: оновлення списку існуючих файлів і видалення файлу з відповідної директорії.

11 Потік подій ВВ Видалення файлів із вхідними даними (11 Видалення всіх файлів)

Передумови: ВВ Видалити файл (9 Видалити файл) виконаний.

Процес: користувач натискає на кнопку видалення всіх файлів, всі існуючі файли в активній на момент роботи програми видаляються.

Післяумови: у папці користувача немає папки з раніше створеними файлами даних, введених з клавіатури.

12 Потік подій ВВ Перевірка поточного ФС (12 Визначення ФС)

Передумови: ВВ Головна сторінка(1 Головна сторінка) виконаний, натиснута кнопка «Перевірити ФС», ВВ (5 Обрати подію) виконаний.

Процес: користувач контролює слово та натискає на кнопку «Визначити ФС», створюється та навчається нейронна мережа Кохонена на даних, які утворені подією

ВВ Персональне налаштування системи (6 Навчання НМ), та проводиться симуляція нейронної мережі з нововведеними з клавіатури даними. Результат симуляції записується до результуючого файлу, виводиться повідомлення про поточний стан оператора та записується до файлу зі статистичними даними.

Післяумови: результат записано в числовому вигляді в текстовий файл, виведено на екран повідомлення про поточний стан та записано до файлу «Статистика» поточний стан та час проходження перевірки.

13 Потік подій ВВ Оновити статистику (13 Оновити статистику)

Передумови: ВВ Перевірка поточного ФС (12 Визначення ФС) виконаний.

Процес: користувач натискає на кнопку «Оновити статистику», відбувається видалення файлу зі статистичними даними користувача.

Післяумови: файл «Статистика» видалено.

14 Потік подій ВВ Перегляд статистику (14 Перегляд статистику)

Передумови: ВВ Перевірка поточного ФС (12 Визначення ФС) виконаний.

Процес: користувач натискає на кнопку «Перегляд статистику», відкриється Excel-файл із статистичними даними, а саме часом проходження перевірки, результату, який отримав користувач та слово, за яким його було перевірено.

Післяумови: Excel-файл відкритий.

На рис. 3.6 представлена діаграма ВВ.

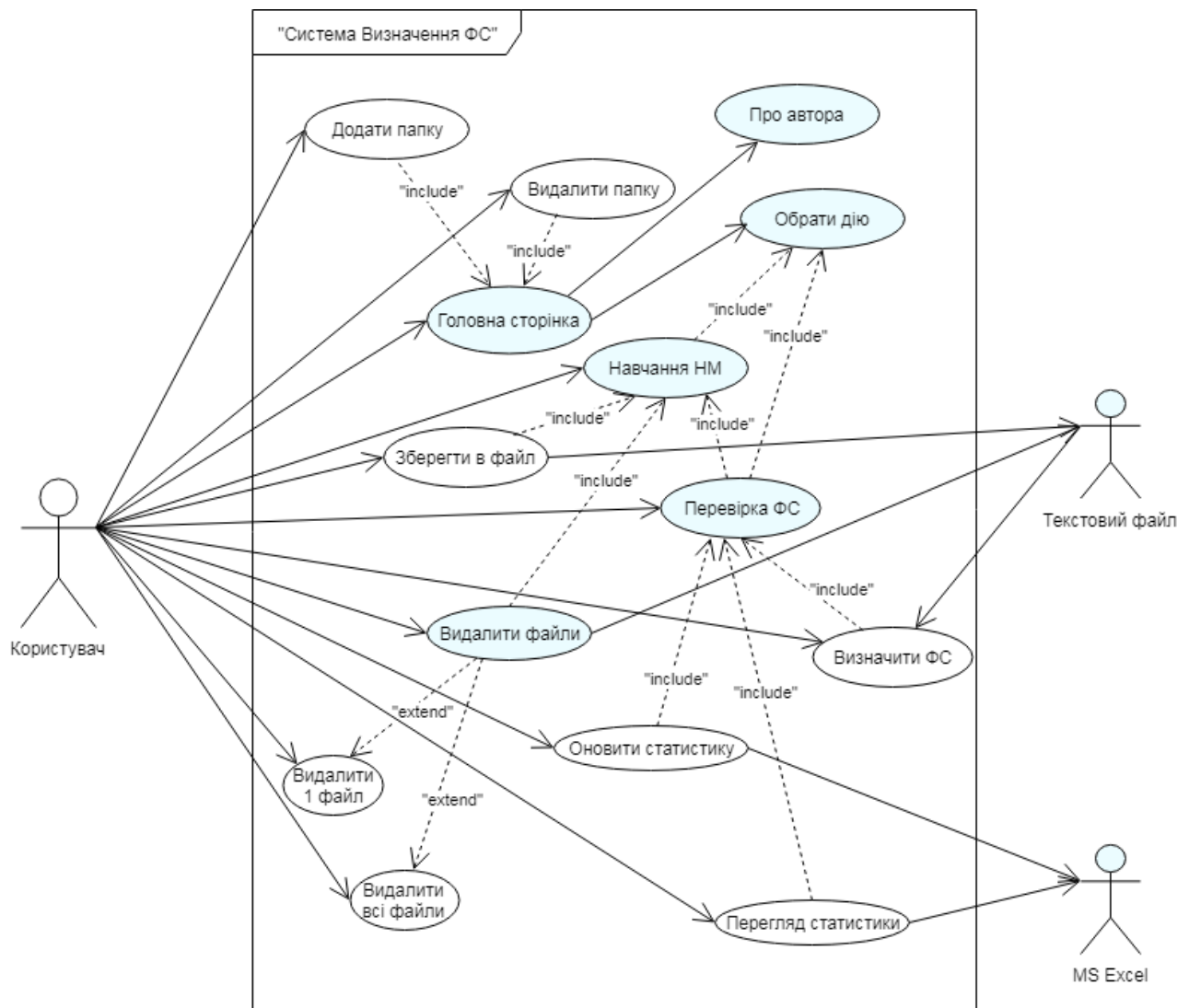


Рисунок 3.6 – Діаграма варіантів використання

3.3 Математична модель для оцінки функціонального стану операторів

Розроблена інформаційна технологія складається з 4 основних етапів (рис. 3.7).

Перший етап. Формування навчальних даних.

Цей етап включає в себе дані з 10 файлів, раніше створених користувачем, де зберігається інформація про введення 10 слів у n-кількість разів. Зчитувалися значення часу натискання та утримання клавіш кожного слова окремо та розділялися по групам відповідно.

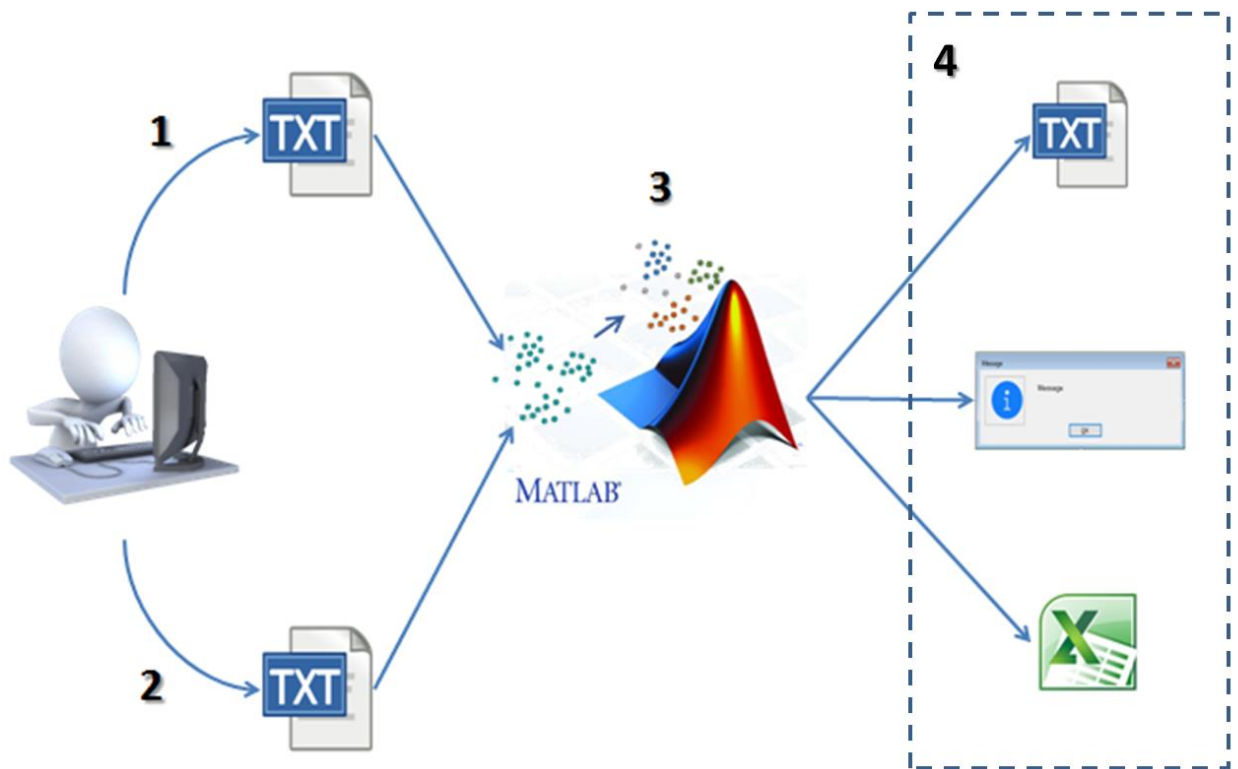


Рисунок 3.7 – Схема основної роботи програмного продукту

Другий етап. Введення контрольного слова.

Користувач (оператор) вводить запропоноване йому слово з 10 літер. Воно зберігається до текстового файлу поточного введення оператора як набір 20 чисел, тобто інформацію про час натискання та час утримання клавіші в мілісекундах.

Третій етап. Вирішення задачі кластеризації.

Робота відбувається в середовищі Matlab. Дані, зібрані на першому етапі, передаються для створення навчальної вибірки. Далі створюється та навчається самоорганізована мапа Кохонена на основі введених значень. Після навчання відбувається внесення нових даних з клавіатури, що описувалися в другому етапі. Результатом задачі кластеризації є вивід значення після симуляції нейронної мережі, до якого саме кластеру віднести дані другого етапу.

Четвертий етап. Формування результату оцінки функціонального стану оператора інформаційних систем.

Значення третього етапу записується до текстового файлу («0» у випадку задовільного стану та «1» - незадовільного стану). Також отриманий результат

впливає у вигляді інформаційного повідомлення для зручного перегляду поточного стану та зберігається до файлу зі статистичними даними, тобто даними про час проходженням, його результатом та словом, на якому перевірявся поточний стан оператора інформаційних систем.

На рис. 3.8 зображено узагальнений алгоритм працюючої схеми.

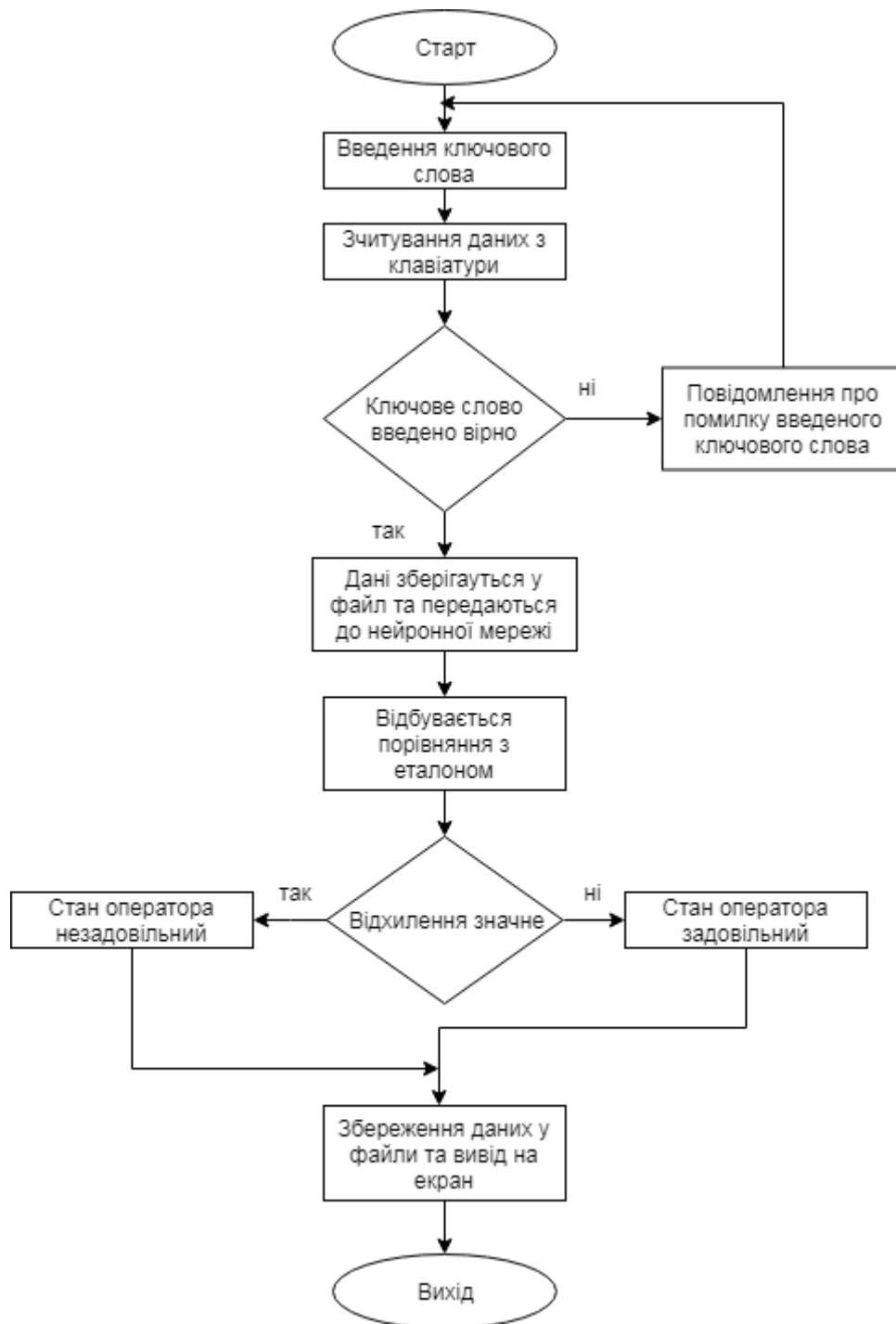


Рисунок 3.8 – Узагальнений алгоритм оцінки функціонального стану людини-оператора інформаційних систем

При розробці математичної моделі оцінки функціонального стану операторів враховано психофізіологічний стан людини. Так як інформаційна технологія розроблена на концепції «чорної скриньки», тому повністю взяти за основу психофізіологічний стан людини неможливо. Отже, взято тільки деякі вегетативні ознаки: гальмування підсвідомих рухів, кров'яний тиск та дихання.

На основі цього було обрано і метод аналізу клавіатурним почерком, адже клавіатура вважається пристроєм введення інформації та обробка її даних може відбуватися саме біометричними методами.

Вхідна інформація для математичної моделі є дані з клавіш клавіатури. Тому для визначення психофізіологічного стану операторів доцільно використати формулу:

$$\Delta t = t_{c-1} - t_c , \quad (3.1)$$

де Δt – час утримання клавіші або час між натисканнями клавіш;

t_{c-1} – тривалість попереднього утримання клавіші або між натисканням клавіш;

t_c – час останнього утримання клавіші або час між натисканнями клавіш (c – поточне значення).

Саме за формулою (3.1) відбувається запис зчитаних даних з клавіатури до файлів для подальшої їх передачі до нейронної мережі. Архітектура штучної нейронної мережі здатна обробити біометричні дані, вирішивши фільтрацію та кластеризацію надійшовши даних з клавіатури.

Нейронна мережа отримує лише ті значення, які зможе обробити та видати більш точний результат. Етап отримання вхідних значень на прикладі двох послідовних натискань клавіш зображено на рис. 3.9.

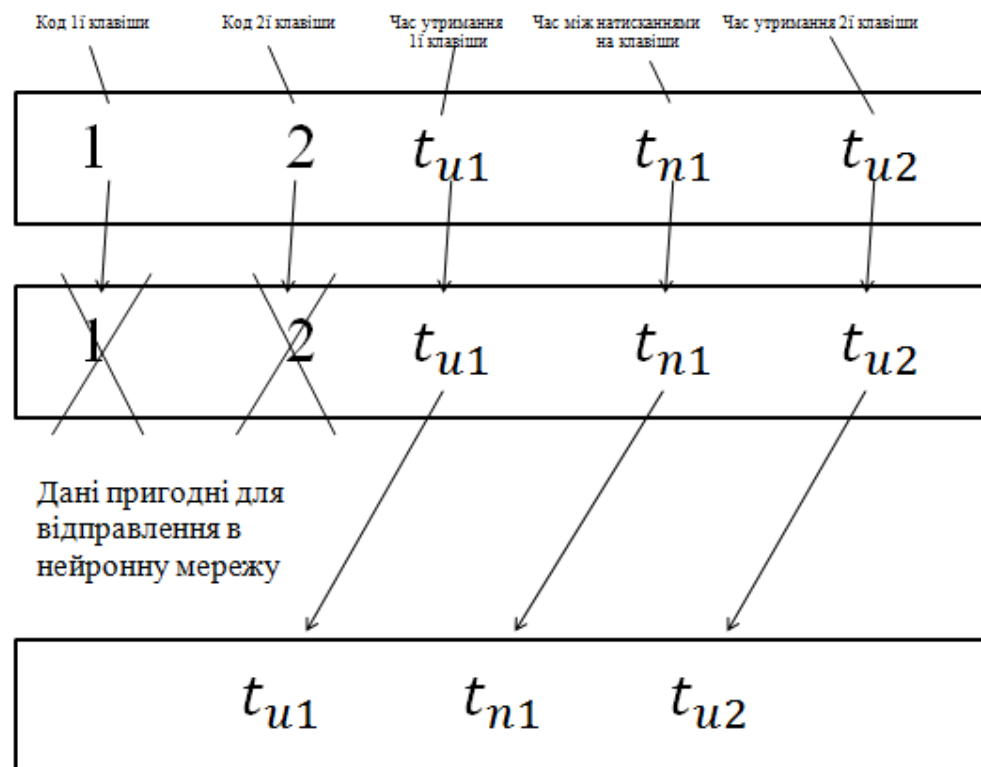


Рисунок 3.9 – Послідовність отримання вхідних значень для створення навчальної вибірки

Основним завданням нейронної мережі для вирішення завдання оцінки функціонального стану операторів інформаційних систем є кластеризація. Попередньо в другому розділі розглянуто існуючі мережі без вчителя та обрано оптимальну мережу – Самоорганізовану мапу Кохонена (SOM) за явні її переваги.

Створення, навчання та симуляція відбувається за допомогою розробленого програмного продукту, який у потрібний момент, а саме визначенні поточного функціонального стану оператора, підключає всі можливі функції Matlab для роботи з нейронними мережами.

Алгоритм навчання SOM [24] має семи обов'язкових кроків для вирішення задачі кластеризації:

1. Визначити значення параметрів алгоритму:
 - метрики, що використовуються при роботі з нейронною мережею;
 - функцію швидкості навчання мережі;
 - функцію, яка зменшує кількість сусідів при збільшенні ітерацій;

- номер ітерацій рівний 0.
- 2. Ініціалізація векторів навчальних ваг нейронів

$$\vec{\omega}^{(j)}(0) \forall j = 1, \dots, M, \quad (3.2)$$

де $\vec{\omega}^{(j)}(0)$ – вектор вагових коефіцієнтів для j -го нейрона з нульовим номером ітерації;

j – індекс;

$M = a \times b$ – загальна кількість нейронів в шарі Кохонена (a – висота, b – ширина шару Кохонена).

3. Перевірка на наявність елементів в навчальній вибірці, Якщо вони є, тоді слід витягнути випадковим чином вхідний вектор, інакше алгоритм припиняється. Вхідний вектор:

$$\vec{x}^{(i)}, N := N - 1, \chi = \chi \setminus \vec{x}^{(i)}, \quad (3.3)$$

де $\vec{x}^{(i)}$ – i -й вектор, який підлягає кластеризації;

$N = |\chi|$ – розмір навчальної вибірки;

$\chi \subset R^n$ – розмір навчальної вибірки.

4. Для $\vec{x}^{(i)}(t)$ та $\vec{\omega}^{(j)}(t) \forall j = 1, \dots, M$ знаходимо густину слою Кохонена:

$$\rho \left(\vec{x}^{(i)}(t), \vec{\omega}^{(j)}(t) \right) = \left\| \vec{x}^{(i)}(t) - \vec{\omega}^{(j)}(t) \right\|^2, \quad (3.4)$$

де $\vec{x}^{(i)}(t)$ – i -й вектор, який підлягає кластеризації;

$\vec{\omega}^{(j)}(t)$ – вектор вагових коефіцієнтів для j -го нейрона.

5. Знаходимо нейрон-переможець ne_c , який лежить близько до поточного об'єкту метрики, яку розглянуто:

$$c = \arg \min \rho(\vec{x}^{(i)}(t), \vec{\omega}^{(j)}(t)), \quad (3.5)$$

$$\forall j \in \{1, \dots, M\}$$

де $\rho(\vec{x}^{(i)}(t), \vec{\omega}^{(j)}(t))$ – мінімальна густина слою Кохонена;

$\vec{x}^{(i)}(t)$ – і-й вектор, який підлягає кластеризації;

$\vec{\omega}^{(j)}(t)$ – вектор вагових коефіцієнтів для j-го нейрона.

6. Для всіх нейронів pe_j виконується розрахунок нових вагових векторів:

$$\vec{\omega}^{(j)}(t+1) := \vec{\omega}^{(j)}(t) + \alpha(t)h_{ci}(t) \left[\vec{x}^{(i)}(t) - \vec{\omega}^{(j)}(t) \right], \quad (3.6)$$

де $h_{ci}(t)$ – функція сусідства між нейронами з індексами c та i на ітерації t ;

$\vec{x}^{(i)}(t)$ – і-й вектор, який підлягає кластеризації;

$\vec{\omega}^{(j)}(t)$ – вектор вагових коефіцієнтів для j-го нейрона.

7. Збільшення ітерації t на 1.

8. Перехід до 3 кроку.

Так як створення і навчання нейронної мережі відбувається в середовищі Matlab слід звернути увагу на функції, якими користується інформаційна технологія для визначення оцінки функціонального стану оператора.

Для створення Самоорганізованої мережі Кохонена використано функцію NEWSOM. Програмно було передано рядок до середовища командного вікна:

```
net = newsom(P,[1 2]);
```

де P – навчальна вибірка;

net – змінна для створення самоорганізованої мережі Кохонена з двома кластерами.

Для навчання – функція TRAIN:

```
net = train(net, P);
```

де net – змінна для тренування самоорганізованої мережі Кохонена.

4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ОЦІНКИ ФУНКЦІОНАЛЬНОГО СТАНУ ОПЕРАТОРІВ

4.1 Результат реалізації інформаційної системи

Для початку роботи з програмою слід встановити її на персональний комп'ютер у будь-якому бажаному для користувача місці. За замовчування створюється ярлик на Робочому столі оператора для швидкого доступу до програми (рис.4.1).

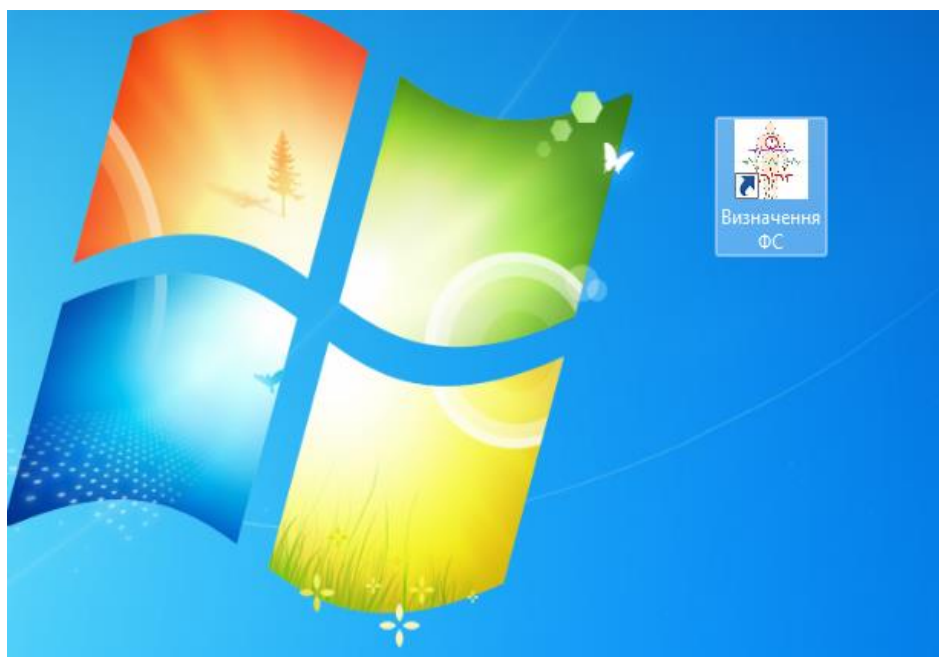


Рисунок 4.1 – Ярлик програми «Визначення ФС» на Робочому столі

Після запуску з'являється головне вікно програми з можливістю вибору (рис. 4.2):

- відомості про автора;
- перегляд існуючих папок;
- додавання нових папок;
- видалення папок.

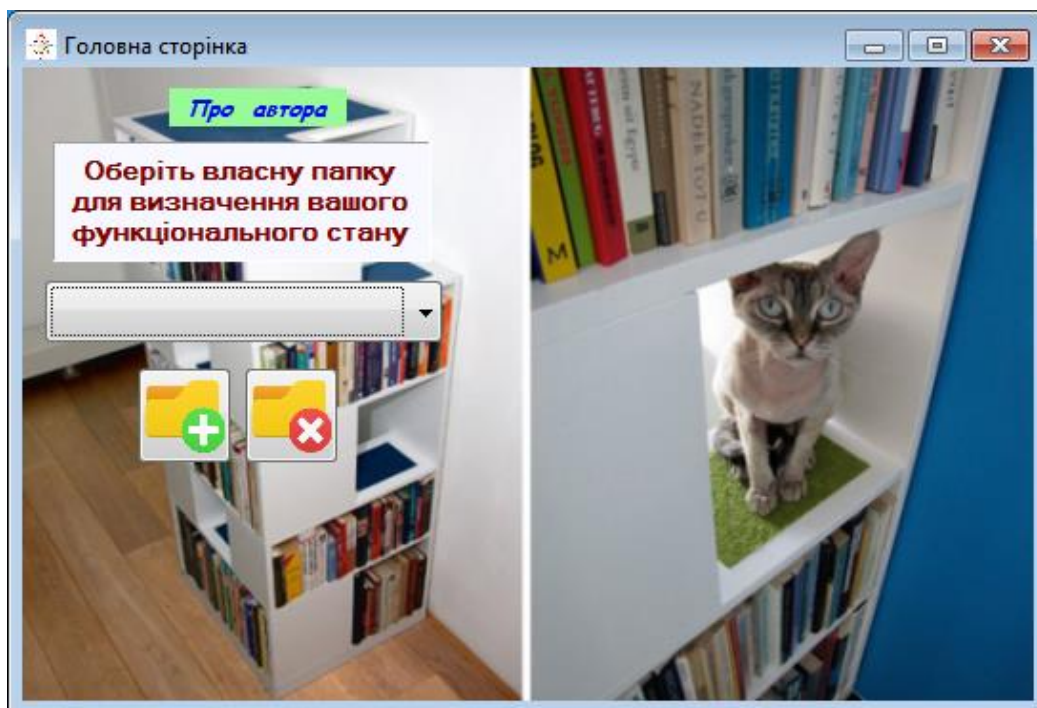


Рисунок 4.2 – Головне вікно програми

Натиснувши на одну з папок з випадного списку, користувачу відкривається вікно з вибором дій: «Навчити під себе НМ» або «Перевірити ФС» (рис. 4.3).

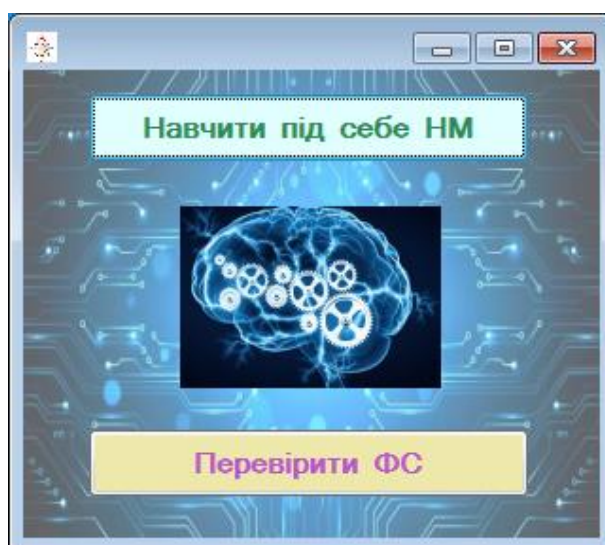


Рисунок 4.3 – Проміжне вікно програми

Обравши першу кнопку користувачу відкривається вікно «Персональне налаштування системи», де задаються вхідні значення до нейронної мережі, тобто створюються навчальні вибірки (рис. 4.4).

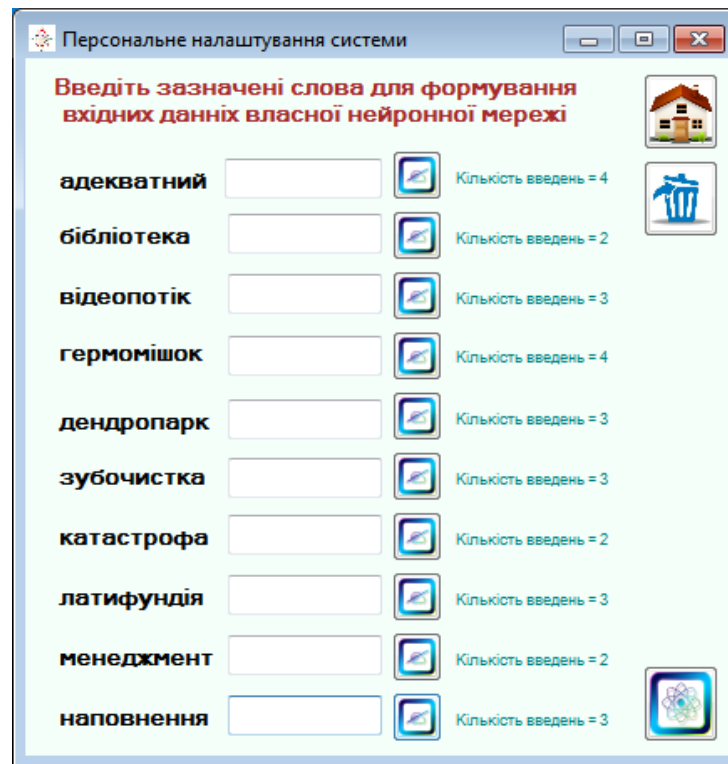


Рисунок 4.4 – Вікно програми «Персональне налаштування системи»

Всі збережені дані знаходяться в директорії диску C, створюються автоматично (рис. 4.5).

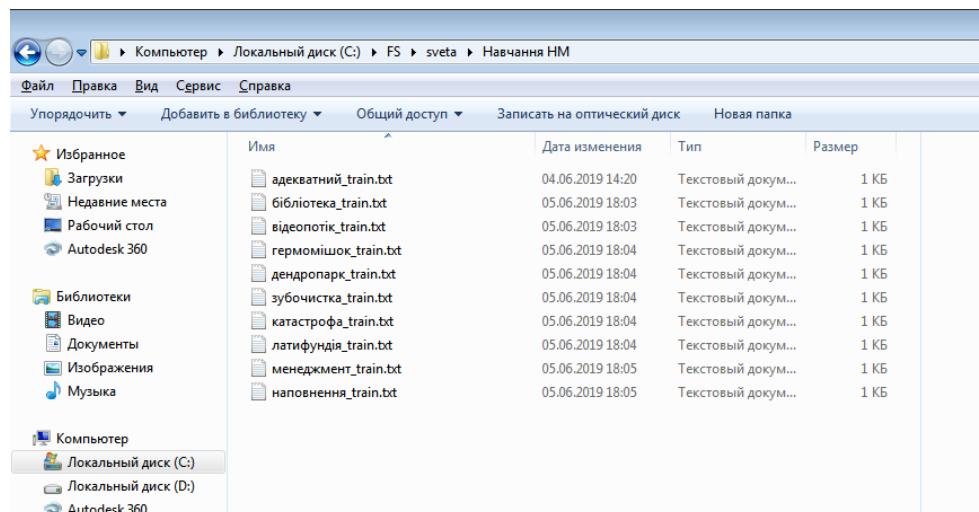


Рисунок 4.5 – Зображення всіх створених файлів

Після створення навчальної вибірки можна видалити за бажання файли, які їх містять, або ж перевірити власний функціональний стан, перейшовши до вікна «Перевірка поточного стану» (рис. 4.6).

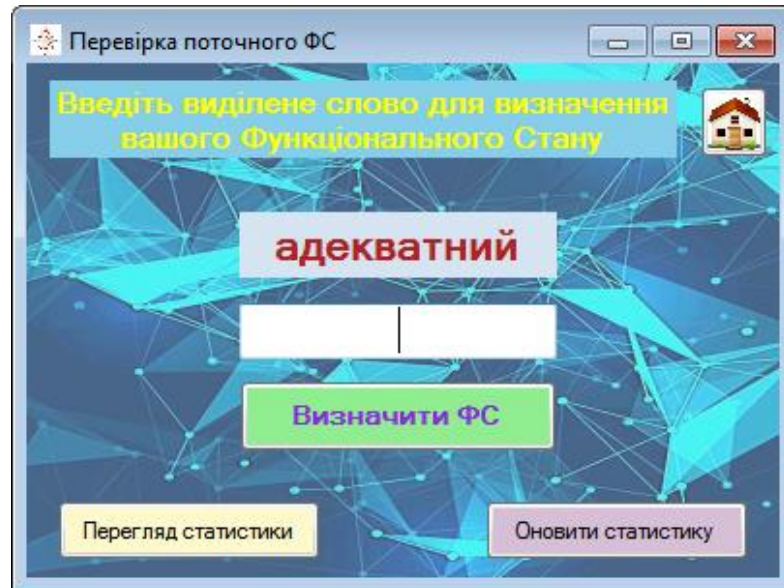


Рисунок 4.6 – Вікно програми, де перевіряється поточний стан користувача

Вводячи запропоноване слово та натискаючи на кнопку «Визначити ФС» йде перевірка оцінки функціонального стану за допомогою нейронної мережі та виводиться результат у вигляді інформаційного повідомлення (рис. 4.7).

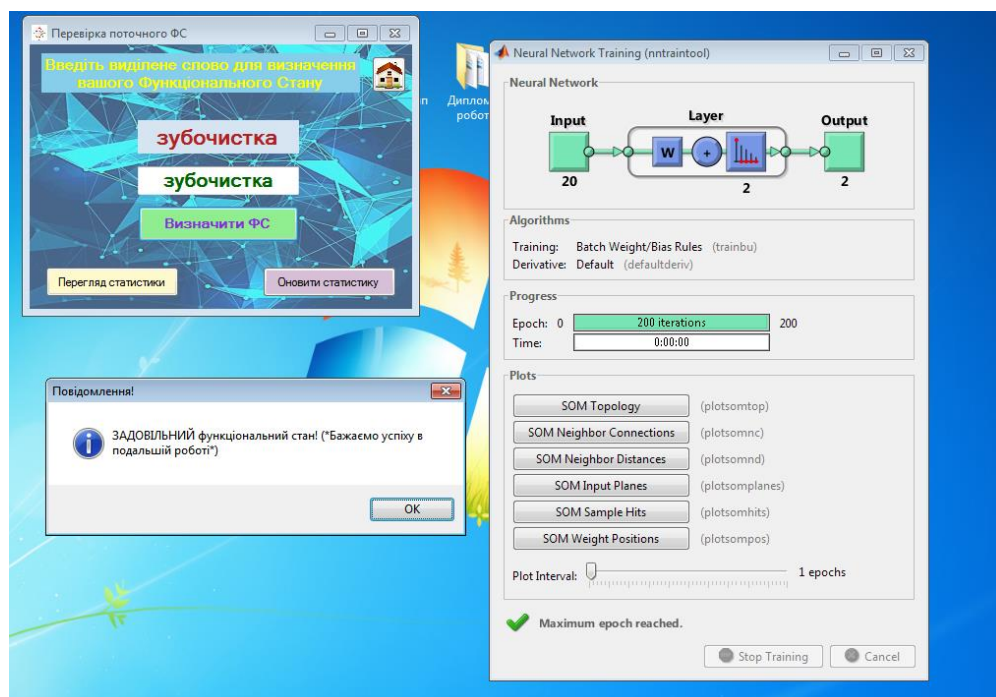


Рисунок 4.7 – Фрагмент робочого столу після визначення функціонального стану

Є можливість переглянути статистичні дані у файлі не виходячи з програми. Слід у вікні «Перевірка поточного ФС» натиснути на кнопку «Перегляд статистики». У результаті відкриється Excel-файл із результатами всіх перевірок певного користувача (рис. 4.8).

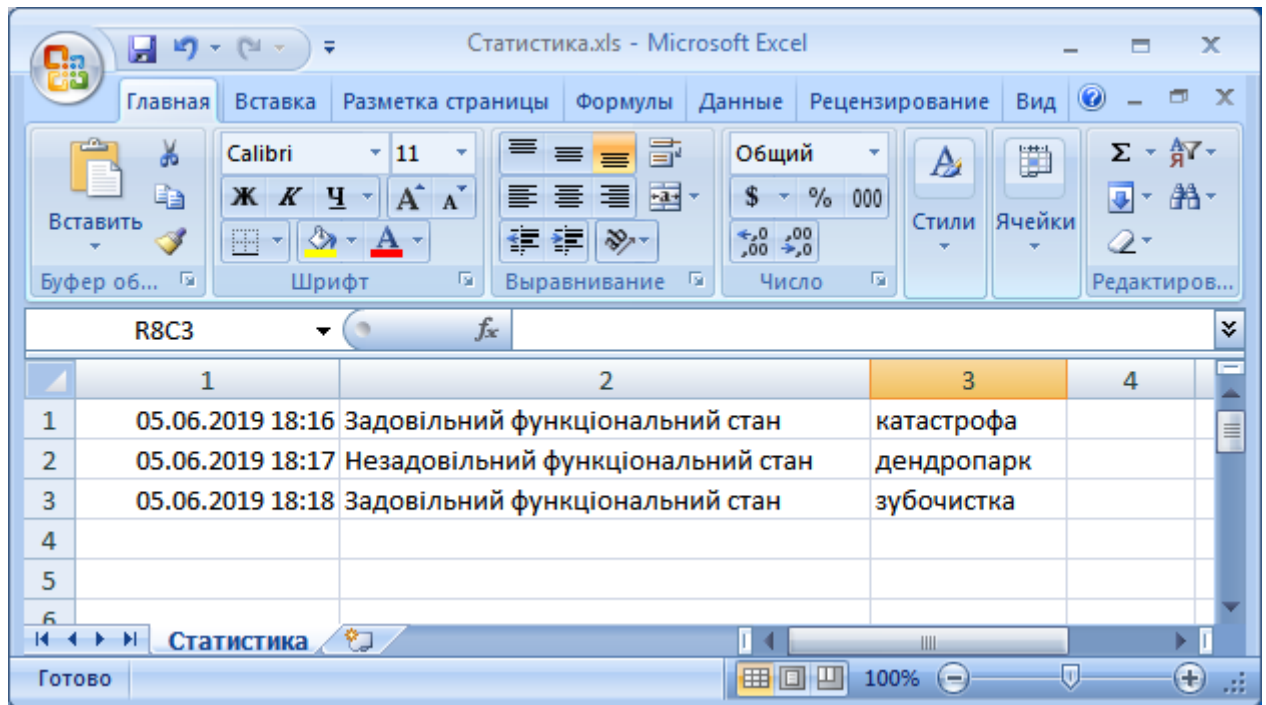


Рисунок 4.8 – Перегляд файлу «Статистика»

У програмі передбачено різні виключні ситуації, всі можливі помилки при роботі з програмою. Коли користувач буде робити ту чи іншу невірну, помилкову дію або ж навпаки створюватиме, додаватиме, то отримуватиме інформаційні повідомлення різного характеру. Також передбачено спливаючі підказки, адже більшість кнопок немає написів (рис. 4.9).

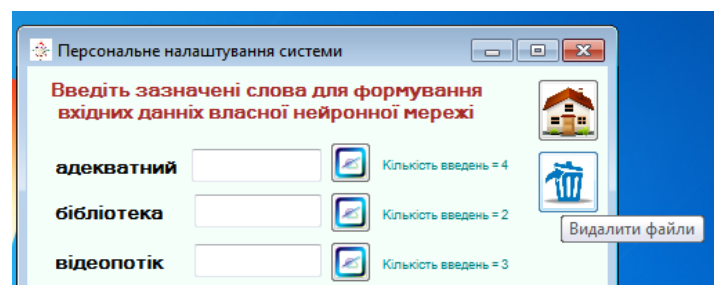


Рисунок 4.9 – Фрагмент одного з прикладів спливаючих підказок

4.2 Проведення експериментальних досліджень

Для підтвердження точності роботи програми оцінки функціонального стану людини-оператора інформаційних систем проведено ряд експериментальних досліджень.

Експерименти проводилися впродовж одного робочого дня за допомогою 5-ти людей-операторів, які проходили тестування власного функціонального стану оцінюючи його двома критеріями.

1. Оцінка за думкою користувача програми власного функціонального стану.
2. Оцінка, яку отримали після проходження перевірки програмою «Визначення ФС».

Оператори мали пройти перевірку кожної робочої години. Тому в результаті вийшло, що у кожного з операторів результатів мало бути 10 власноруч відміченими та 10 визначеними машиною, що записувалися до файлу «Статистика». Для зручності подальшого дослідження експериментів користувачі стави «1», якщо психофізичний стан задовільний, та «0» - незадовільний.

Для побудову графіків точності програмного продукту було створено таблицю в MS Excel з даними про щогодинний поточний стан 5-ти операторів (рис. 4.9).

	A	B	C	D	E	F	G	H	I	J	K
1	Години	Оператор 1	Оператор 2	Оператор 3	Оператор 4	Оператор 5					
2	8:00	0	0	1	1	1	1	1	0	1	1
3	9:00	1	1	1	1	1	0	1	1	1	1
4	10:00	1	0	1	1	1	0	1	1	0	1
5	11:00	1	1	1	1	1	1	0	1	1	1
6	12:00	0	0	0	0	1	1	0	0	1	1
7	13:00	1	1	1	1	1	1	1	1	1	1
8	14:00	1	0	1	1	1	1	1	1	1	1
9	15:00	1	1	1	0	1	0	0	0	1	1
10	16:00	1	1	1	1	0	0	0	0	1	1
11	17:00	0	1	1	1	0	0	1	1	0	0

Рисунок 4.9 – Зображення таблиці порівнянь людських відчуттів і машинного розуму

У таблиці позначено 5 операторів, що проходили перевірку, та їх результати по власним відчуттям (білий колір стовпця) і по математичній моделі програмного продукту (зелений колір).

За результатами дослідження по кожному з операторів побудовано графіки відповідності людських відчуттів і машинного розуму (рис. 4.10 – 4.15).

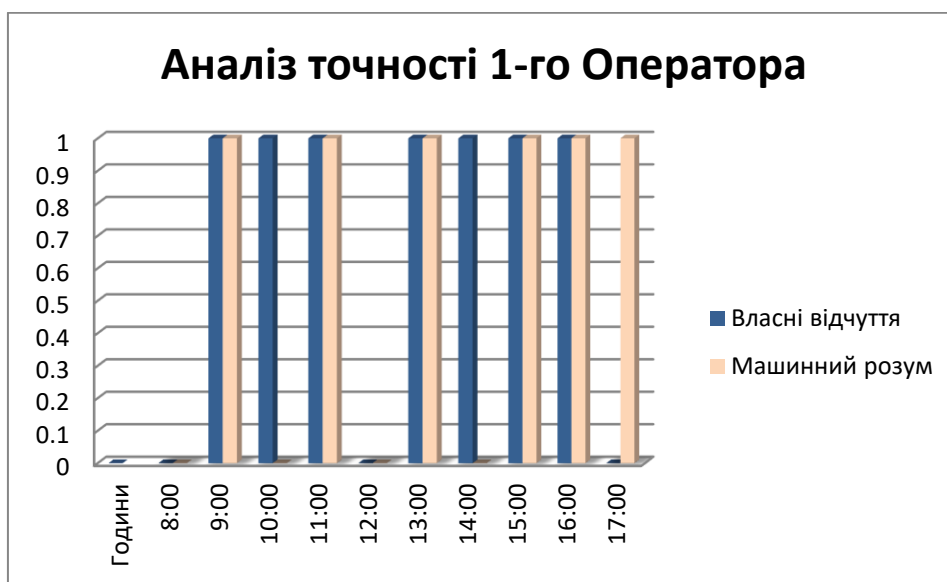


Рисунок 4.10 – Результат порівнянь першого оператора

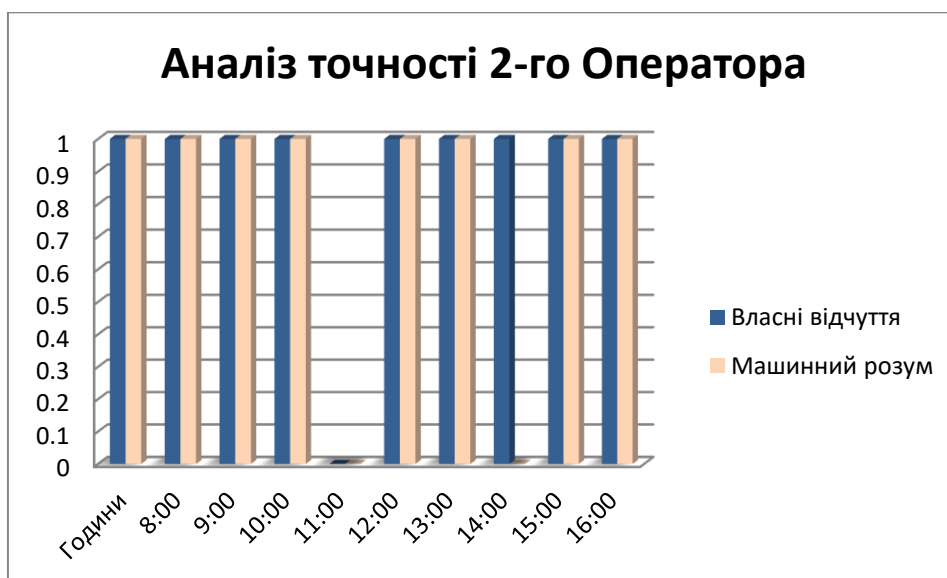


Рисунок 4.11 – Результат порівнянь другого оператора



Рисунок 4.12 – Результат порівнянь третього оператора



Рисунок 4.13 – Результат порівнянь четвертого оператора



Рисунок 4.14 – Результат порівнянь п'ятого оператора

Розрахуємо середнє значення точності методу для кожного з операторів окремо.

$$Correctness = \frac{\sum_{i=1}^n \sum_{j=1}^s \alpha_{ij}}{n \cdot s}, \quad (4.1)$$

де i – номер експерименту, $i = 1, n$; (n – кількість експериментів);

j – номер людини оператора, $j = 1, s$; (s – кількість операторів);

$\alpha_{ij} = \begin{cases} 1, & \text{якщо оцінка програми співпадає з оцінкою оператора} \\ 0, & \text{якщо оцінка програми НЕ співпадає з оцінкою оператора} \end{cases}$

Середнє значення точності для першого оператора = 0,7.

Середнє значення точності для другого оператора = 0,9.

Середнє значення точності для третього оператора = 0,7.

Середнє значення точності для четвертого оператора = 0,8.

Середнє значення точності для п'ятого оператора = 0,9.

Узагальнений графік характеристики точності зображений на рис. 4.15.

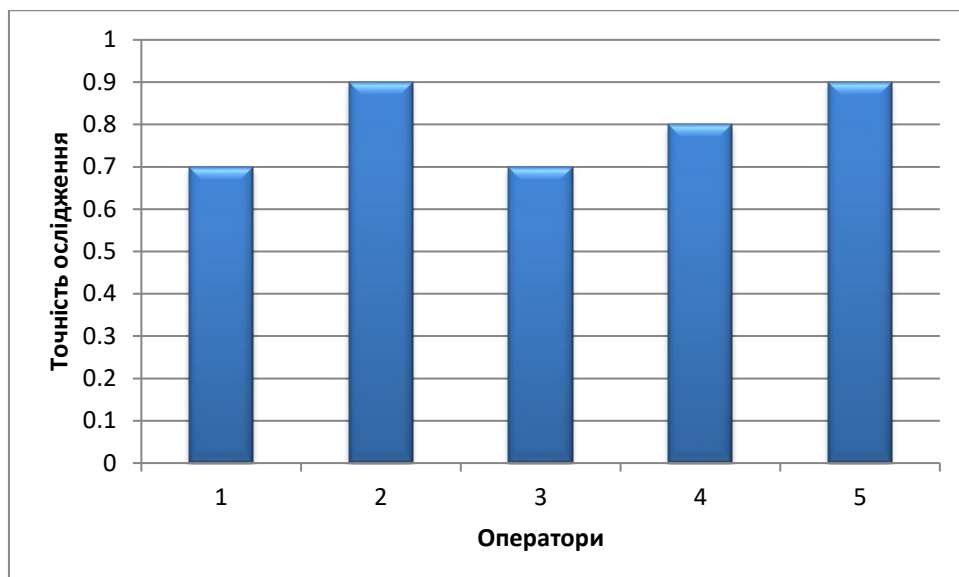


Рисунок 4.15 – Узагальнена характеристика точності

Для розрахунку середнього значення точності методу для програмного продукту «Визначення ФС» слід зібрати до купи всі перевірки точності операторів та загальну суму експериментів.

Проведений дослід показує, що середнє значення точності методу рівне 0,8.

Отже, експериментальні дані підтверджують математичну модель для оцінки функціонального стану операторів інформаційних систем.

ВИСНОВКИ

Задача автоматизації оцінки функціонального стану операторів інформаційних систем широко досліджена, але не має кінцевих результатів в розробці програмних продуктів.

Результатом дипломного проекту є програмний продукт оснований на алгоритмах та математичних моделях, які при проведенні експериментальних досліджень дав високий результат точності.

На відміну від апаратних методів контролю функціонального стану, запропонований метод використовує модель, основу на використанні апарату нейронних мереж, що забезпечує суттєве зростання точності і характеризується невибагливістю до апаратного забезпечення.

У роботі використано метод аналізу клавіатурного почерку, так як він має низку переваг серед інших біометричних методів. Метод використовувався при введенні даних з клавіатури та створенні навчальної вибірки для нейронної мережі.

Серед існуючих нейронних мереж за проведеним аналізом виявлено, що оптимальною мережею без вчителя є Самоорганізована карта Кохонена. Мапа не вибаглива та легко пристосовується до психофізичних навантажень.

У результаті експериментальних досліджень отримано точність в 80 %.

Програмний продукт «Визначення ФС» може бути використана в автоматизованих технологічних комплексах різноманітного призначення і сприяє удосконаленню роботи операторів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лавров Е.А., Скиданенко А.С.//Эргономические резервы повышения эффективности АСУТП производства удобрений. Сучасні інформаційні системи і технології: матеріали Другої міжнародної науково-практичної конференції, м.Суми,21-24 травня 2013 р. / редкол.А.С. Довбиш, О.А. Борисенко, О.В. Бондар. Суми: Сумський державний університет,2013-С.53-54.
2. Ключарев В.А., Никишина И.С., Лысков Е.Б., Сандстрем М., Хансон Милд К. Влияние слабых электромагнитных полей на стабильность изображения компьютерного монитора: возможные последствия для операторов Текст. / Физиология человека. 2000, т.26, сч.54.
3. Иванов А. И. Нейросетевые технологии биометрической аутентификации пользователей открытых систем Текст. // Автореф. дис. . доктора техн. наук. Пенза: Изд-во Пенз. гос. ун-та, 2002.
4. Ярослав Ж. Биометрические методы аутентификации [Электронный ресурс] / Жиронкин Ярослав // Jet Info. 2017. URL: <http://www.jetinfo.ru/stati/biometricheskie-metody-autentifikatsii>.
5. Рудаков О. М. Метод біометричної аутентифікації, заснований на аналізі клавіатурного почерку // Молодий вчений. 2016. №11. С. 448-451.
6. Нейромережа ідентифікація типу особистості людини по клавіатурного почерку. Т.В. Жашкова, О.М. Шарунова, Е. Ш. Ісяново. Міжнародний студентський науковий вісник. 2015. №3 Ч.1. С. 144-146
7. Використання клавіатурного почерку для аутентифікації в розподілених системах з мобільними клієнтами. Соколов Д.А. Безпека інформаційних технологій. 2010 року. № 2. С. 50-53.
8. В.І.Волчіхін, А.І.Іванов Природне використання штучних нейронних мереж в біометрії [Текст]. // "Системи безпеки" №3 2002 р. Стор 48-49.

9. Сидоровкіна І. Г. Три алгоритми управління доступом до КСІІ на основі розпізнавання клавіатурного почерку оператора / І. Г. Сидоровкіна, О. М. Савінов. // Вісник Чувашського Університету. 2013. №3. С. 293–301.
10. Навчання нейронної мережі // AIportal. 2009. URL: <http://www.aiportal.ru/articles/neural-networks/learning-neunet.html>.
11. Neural Network Toolbox Створення, навчання та моделювання нейронних мереж // MathWorks. 2019. URL: https://matlab.ru/products/neural-network-toolbox/neural-network-toolbox_rus_web.pdf.
12. Круглов В. В. Штучні нейронні мережі. Теорія та практика / В. В. Круглов, В. В. Борисов. М: Стереотип, 2002. 382 с. (Гаряча лінія - Телеком).
13. Нейронна мережа Кохонена. URL: <http://inmad.vntu.edu.ua/portal/static/D28C6207-9C44-4C8C-B56E-43E2CF72C372.pdf>.
14. Старіков О. Самоорганізована карта Кохонена - математична модель / Олексій Старіков // Технологія аналізу даних. 2008. URL: <https://basegroup.ru/community/articles/som>.
15. Advantages And Disadvantages of Visual Basic. 2011. URL: <https://iiteeestudents.wordpress.com/2011/08/28/advantages-and-disadvantages-of-visual-basic/>.
16. Neural Network Toolbox. 2014. URL: <http://matlab.exponenta.ru/neuralnetwork/index.php>.
17. Demuth H. Neural Network Toolbox / H. Demuth, M. Beale., 2004. 141 с. (MathWorks).
18. Raj Kumar Bansal. MATLAB and its applications in engineering : (based on MATLAB 7.5 (R2007b)) / Raj Kumar Bansal, Ashok Kumar Goel, Manoj Kumar Sharma. India: Person Education, 2008. 520 с.
19. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем Интернет-университет информационных технологий 2-е изд. М.: Бином. Лаборатория знаний Интуит Серия: Основы информационных технологий, 2008. 300 с.

20. Марка Д.А., МакГоун К. Методологія стурктурного аналізу та проєтування SADT: Пер. з англ. С.: Метотехнологія, 2003.
21. SADT, A Dictionary of Computing // Oxford University Press. 2016. URL: <https://www.encyclopedia.com/science-and-technology/computers-and-electrical-engineering/computers-and-computing/sadt>.
22. IDEF0 (Integration Definition for Function Modeling) // BCG. 2017. URL: http://b-c-group.ru/?page_id=188.
23. Use Case // Techopedia. 2019. URL: <https://www.techopedia.com/definition/25813/use-case>.
24. Самоорганізуюча карта Кохонена (алгоритм кластеризації) // algowiki. 2016. URL: [https://algowiki-project.org/ru/Самоорганізуюча_карта_Кохонена_\(алгоритм_кластеризації\)](https://algowiki-project.org/ru/Самоорганізуюча_карта_Кохонена_(алгоритм_кластеризації)).
25. Тимошук П. В. Імпульсна нейронна мережа типу "K-winners-take-all" [Текст] / П. В. Тимошук // Вісник Національного університету "Львівська політехніка". Автоматика, вимірювання та керування. 2017. № 880. С. 41-50.
26. Создание модели данных с помощью AllFusion ERwin Data Modeler [Текст]: метод. указ. / сост.: Ю.А. Соловьева, Н.Б.Соколова. Новокузнецк : СибГИУ, 2009. 32 с.
27. Altaf S. Implementation of Efficient Artificial Neural Network Data Fusion Classification Technique for Induction Motor Fault Detection [Текст] = Впровадження ефективної методики класифікації злиття даних для визначення несправностей асинхронного двигуна із застосуванням штучної нейронної мережі / S. Altaf, M. S. Mehmood, M. Imran // Журнал інженерних наук. 2018. Т. 5, № 2. С. E16-E21.
28. Kuzik O. V. Neural Networks and Artificial Intelligence [Текст] / O. V. Kuzik, M. M. Glazunov // Інженерія програмного забезпечення. 2010. № 3. С. 68-71.
29. Акчурін О. М. Функціональний стан вегетативної нервової системи при гострому та хронічному пієлонефриті у дітей старшого шкільного віку [Текст]: автореферат... канд. мед. наук, спец.: 14.01.10 педіатрія / Акчурін О. М. Х.: Харківська мед. акад. післядиплом. освіти, 2007. 20 с.

30. Лавров Е. А. Подход к созданию интеллектуального агента для системы эргономического обеспечения электронного обучения [Текст] / Е. А. Лавров, Н. Л. Барченко // Сучасні інформаційні системи і технології: матеріали Другої міжнародної науково-практичної конференції, м. Суми, 21-24 травня 2013 р. / Ред.кол.: А.С. Довбиш, О.А. Борисенко, О.В. Бондар. Суми : СумДУ, 2013. С. 93-94.

31. Лавров, Е.А., Барченко, Н.Л. Подход к выбору типа диалога для адаптивных обучающих систем «человек-компьютер» на основе анализа предпочтений оператора [Текст]/ Е.А. Лавров, Н.Л. Барченко // Восточно-Европейский журнал передовых технологий. Сер. Системы управления. Харьков, 2009 3/4 (39) С. 45-49.

32. Абашин В.Г. Адаптивная математическая модель мультибио-метрической подсистемы определения работоспособности человека-оператора АРМ на основе нечетких множеств. Информационные системы и технологии 2011. №5(67).С. 90-96

33. Чабаненко, П. П. Исследование безопасности и эффективности функционирования систем человек-техника эргосетями [Текст]/П.П. Чабаненко Акад. воен.-мор. сил им. П. С. Нахимова. Севастополь: Изд-во Акад. воен.-мор. сил им. П. С. Нахимова, 2012. 160 с.

34. Aveal. Самоорганизующиеся карты Кохонена. Часть 1. / Aveal // MicroTechnics. 2015. URL: <https://microtechnics.ru/samoorganizuyushhiesya-karty-koxonena-chast-1/>.

35. Працездатність, функціональні стани та ефективність діяльності людини-оператора // StudFiles. 2016. URL: <https://studfiles.net/preview/5188884/page:64/>.

36. Matlab Installation procedure // SD Pro Engineering Solutions Pvt Ltd. 2014. URL: <https://www.youtube.com/watch?v=krEPOXPvNIg>.

37. Лавров Є.А., Барченко Н.Л., Рудакова Н.О., Вакал С.М. Комп'ютерна технологія прогнозування якості навчальної діяльності в модульних системах електронного навчання // Інформатика, Математика, Автоматика: матеріали

науково-технічної конференції, м. Суми, 05-09 лютого 2018р. Суми : Сумський державний університет, 2018. С.117.

38. Лавров Є.А., Барченко Н.Л., Вакал С.М. Забезпечення когнітивного комфорту в E-learning // Цифрові технології в освіті, науці, суспільстві: матеріали XII міжнародної науково-практичної конференції (4-6 грудня 2018 року), Петрозаводск, О.Ю. Насадкіна (відп. редактор).2018. Петрозаводск, 2018. с. 144 – 147.

39. Лавров Є.А., Вакал С.М. Інформаційна технологія оцінки функціонального стану операторів інформаційних систем // Інформатика, Математика, Автоматика ІМА::2019 : матеріали та програма науково-технічної конференції, м. Суми, 23-26 квітня 2019 р. Суми : Сумський державний університет, 2019. С.91.

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

**на розробку інформаційної системи оцінки функціонального стану операторів
інформаційних систем**

Призначення інформаційної системи

Інформаційна система (ІС) призначена автоматизувати визначення оцінки функціонального стану людей-операторів інформаційних систем.

Мета створення інформаційної системи –

Розробити інформаційну систему, яка б давала якісні результати. Спрямувати програму на удосконалення роботи операторів та автоматизацію оцінки їх функціонального стану

Цільова аудиторія

У цільовій аудиторії ІС можна виділити наступні групи:

1. Студенти.
2. Тестувальники.
3. Оператори інформаційних систем.
4. Інші зацікавлені сторони.

Вимоги до інформаційної підсистеми

Розроблювана система повинна бути додатком, клієнтська частина якого є windows-формою з набором елементів керування, які забезпечать виконання функціональних можливостей, визначених у пункті «Вимоги до функціональних характеристик».

Вимоги до технологій розробки

Система розробляється ітеративно із врахуванням принципів та технологій уніфікованого процесу розроблення програмного забезпечення.

Програма повинна бути написана мовою Visual Basic із використанням технології Microsoft .Net та доступом до середовища Matlab.

Вимоги до інформаційної та програмної сумісності

Інформаційна система повинен мати інсталяційний пакет для встановлення на ОС Windows. Для коректної роботи додатку необхідний .Net Framework 4.5 або вище. Обов'язково встановлене програмне забезпечення Matlab, починаючи з версії R2011b та вище.

Інтерфейс продукту повинен бути оформлений українською мовою.

Система створюється для використання на платформі Windows із будь-яким текстовим редактором, при роздільній здатності екрану не нижче 1024x768 (стандартний масштаб екрану операційної системи 96dpi).

Вимоги до функціональних характеристик

Система повинна забезпечувати виконання наступних функцій, які ініціюються за допомогою меню та натисканням кнопок:

- виведення інформації про автора та короткий огляд функцій програми;
- автоматичне створення папок та необхідних файлів;
- зберігання зчитаних з клавіатури даних в txt-файл:
 - час натискання на клавішу;
 - час утримання клавіши;
- створення еталонного значення за результатами зчитаних даних в середовищі Matlab;

- порівняння створеного еталонного значення з поточно-введеними даними в середовищі Matlab;
- виведення результату порівняння на екран у вигляді повідомлення та збереження результату в txt-файл та Excel-файл;
- додавання та видалення утворених папок та файлів у рамках системи.

Короткий опис головного потоку виконання системи

Для початку роботи програми слід встановити її на комп'ютер. Після запуску відкривається головне вікно програми, де розташовано ряд кнопок:

- Додавання папки (значок, що відкриває приховану панель із текстовим полем та кнопкою «Додати папку»);
- Видалення папки (значок, що відкриває приховану панель із випадним списком папок та кнопкою «Видалити папку»);
- Дані про автора;
- Випадний список папок, при натисканні на який відбувається перехід до проміжного вікна заданої дерик торії.

Потрапивши до проміжного вікна оператор має змогу натиснути дві кнопки «Навчити під себе НМ» або ж «Перевірити ФС».

Перейшовши до вікна «Персональне налаштування системи», було обрано кнопку «Навчити під себе НМ». На формі знаходиться ряд функцій:

- Перехід на головну сторінку;
- Введення та додавання слів в файли для формування навчальної вибірки;
- Видалення файлів або конкретного файлу;
- Перехід до перевірки поточного функціонального стану.

Перехід по кнопці «Перевірити ФС» відбувається лише в тому випадку, коли навчальна вибірка сформована, тобто хоча б раз відбулося введення та збереження в файли певної директорії, що прописано програмно, контрольних слів.

Значок видалення файлів або конкретного файлу відкриває нову форму, де користувач обирає, що саме він хоче видалити. Якщо це один файл, то з випадного списку слід обрати один файл та натиснути на значок видалення, якщо ж всі, тоді на значок видалити все.

Перейшовши у форму визначення поточного функціонального стану оператора, користувач бачить:

- Активне текстове поле, в яке необхідно ввести контрольне слово зазначене вище;
- Кнопка «Визначити ФС»;
- Кнопка «Переглянути статистику»;
- Кнопка «Оновити статистику».

Ввівши слово та натиснувши кнопку «Визначити ФС» запускається частина середовища Matlab, а саме nntool, де відбувається візуалізація виконуваної дії. Тобто за допомогою системи передаються попередньо введені дані для створення та навчання нейронної мережі та поточно-введені дані з клавіатури для перевірки, до якого з кластерів значення більше входять.

У результаті оператор отримує можливість відстеження кожного введеного значення, його приналежність до кластерів на вікні навчання та на екрані інформаційне повідомлення про поточний функціональний стан («задовільний» та «незадовільний»).

Щодо кнопок «Переглянути статистику» та «Оновити статистику». Перша відкриває файл MS Excel для перегляду всіх збережених даних визначення функціонального стану, час виконання перевірки, результат та слово, за яким була перевірена та чи інша особа.

Кожна дія користувача супроводжується спливаючими підказками або інформаційними повідомленнями.

Вимоги до надійності

Система не повинна вимагати від користувача спеціальної підготовки, не пов'язаної з їх професійними обов'язками. Інформаційна система повинна мати інструкцію користувача.

Програма має забезпечувати перевірку вводу користувачем значень: на всіх формах програми до всіх введених значень і здійснюваних дій застосовуються наступні правила перевірки:

- поле вибору новостворюваної папки: рядок, який складається з літер від А до Я та знак «'»;
- поле вибору ключових слів: рядок, який відповідає вказаному в завданні ключовому слову.

При введенні значення в невірному форматі користувачеві повинне бути показано повідомлення про помилку.

Також слід відмітити, що в програмі є випадні списки, які теж обмежені в використанні. Натискання на список дозволяє переміститися в дану директорію папки, яку вибрав користувач, та при видаленні або папки, або файлу, слід спочатку натиснути на свій вибір із випадного списку, а потім подальшу кнопку, в залежності, яку дію користувач хоче зробити.

Загальні відомості

Система не повинна бути перенавантаженою надлишковою інформацією. Текст повинен бути зручним для читання. Кольори та елементи на сайті не повинні відволікати увагу.

Вміщення головних форм вікна показано на рис. А1 – А3.

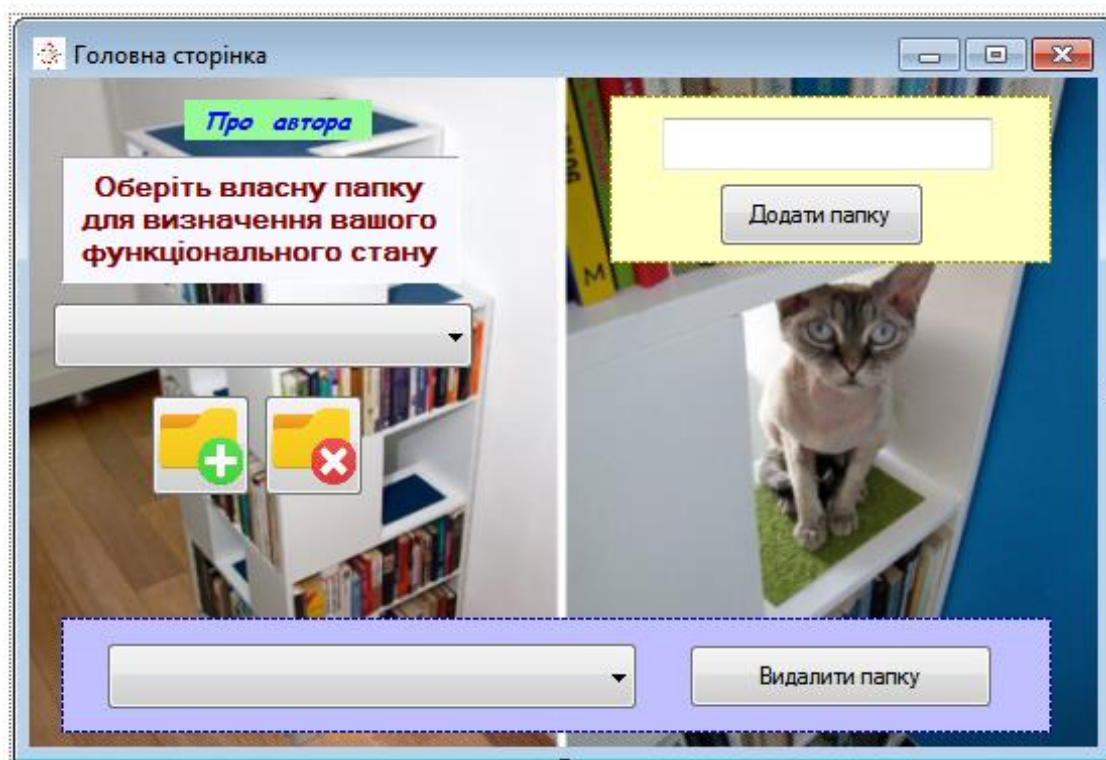


Рисунок А1 – Макет головного вікна програми

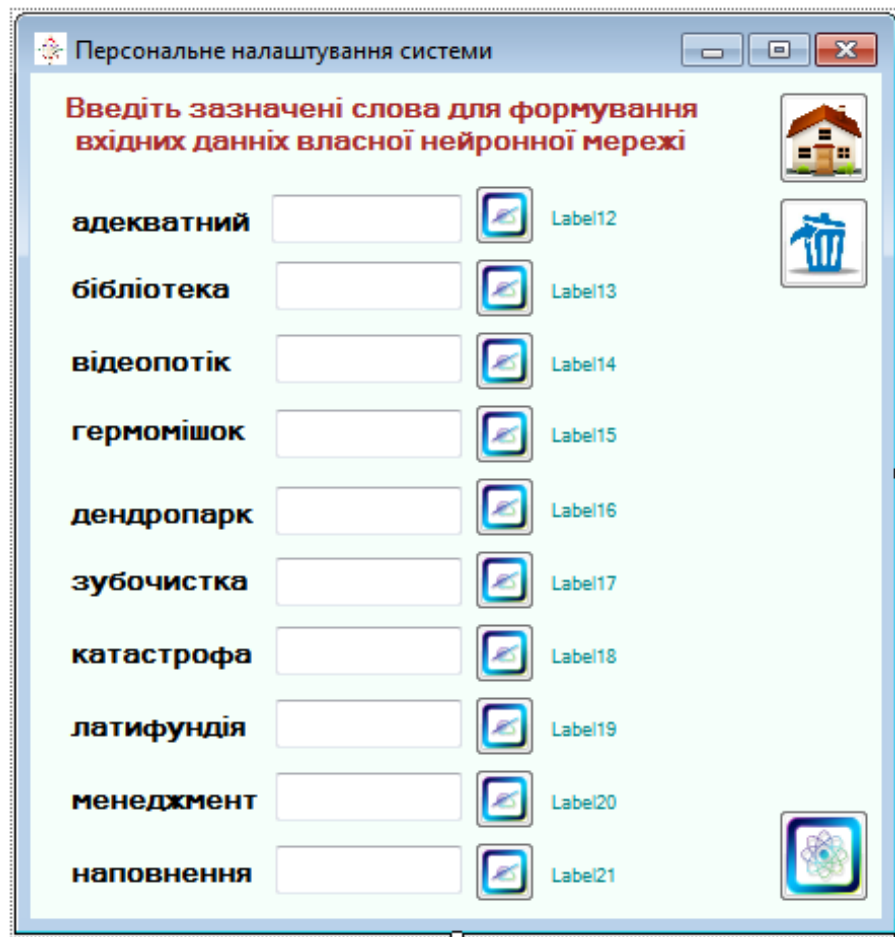


Рисунок А.2 – Макет вікна створення навчальної вибірки для нейронної мережі

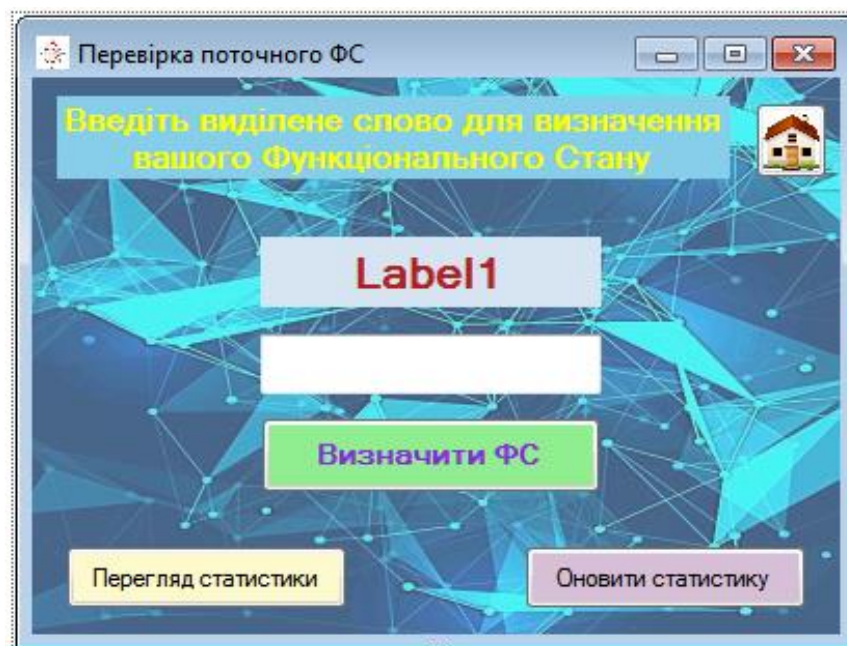


Рисунок А.3 – Макет вікна оцінки функціонального стану оператора інформаційної системи

Склад і зміст робіт зі створення інформаційної підсистеми

Докладний опис етапів роботи зі створення інформаційної системи наведено в табл. А.1.

Таблиця А.1 – Етапи створення інформаційної системи

	Склад і зміст робіт	Строк розробки (у робочих днях)
	<p>Розробка каркасу: Проектування розмітки та наповнення форм інформаційної системи</p>	3 дні
	<p>Введення даних: Реалізація програмного введення даних для формування навчальної вибірки</p>	1 день
	<p>Навчання: Створення еталонного значення за введеними з клавіатури даними</p>	4 дні
	<p>Симуляція: Реалізація порівняння поточних вхідних даних з клавіатури з еталонним значенням</p>	2 дні
	<p>Головний вивід: Реалізація виводу у вигляді повідомлення результату проведення симуляції (визначення функціонального стану оператора) та занесення результату до файлів після натискання на кнопку «Визначити ФС»</p>	3 день
	<p>Завершення роботи: Проведення стилістичних виправлень підсистеми, перевірка (тестування) реалізованого функціоналу</p>	1 дні

Загальна тривалість робіт (з урахуванням резервного строку на налагодження й виправлення помилок) і строк закінчення проекту – 14 днів (тобто майже 3 тижні).

Вимоги до складу й змісту робіт із введення системи в експлуатацію

Для створення умов функціонування, при яких гарантується відповідність створюваної системи вимогам сьогодення технічного забезпечення і можливість його ефективної роботи, в організації Замовника повинен бути проведений певний комплекс заходів.

Для встановлення інформаційної системи на комп'ютер необхідно, щоб параметри забезпечень відповідали вимогам, що зазначені в технічному забезпеченні.

ДОДАТОК Б. Планування робіт

Деталізація мети проекту методом SMART. У ході роботи над проектом необхідно створити інформаційну технологію. Конкретизуємо мету проекту, скориставшись методом SMART, що є однією з найбільш відомих і ефективних методик ефективного планування. П'ять букв у слові SMART позначають п'ять ключових, вимог до ефективного визначення цілі при створенні проектів. Результати деталізації методом SMART розміщені у табл. Б.1.

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити підсистему для ідентифікації функціонального стану людини – оператора за допомогою клавіатурного почерку.
Measurable (вимірювана)	Оцінювання функціонального стану оператора визначити за рахунок впливу психофізіологічного стану людини на клавіатурний почерк.
Achievable (досяжна)	Поставлена мета є реальною, тому що є системи аналоги, які використовуються в інших галузях. Тому для досягнення поставлених цілей потрібно проаналізувати існуючі системи.
Relevant (реалістична)	Для досягнення поставленої мети є всі програмні, технічні та емпіричні знання, тому ціль є реалістичною.
Time-framed (обмежена у часі)	Ціль обмежена у часі, тому що результатів оцінки функціонального стану необхідно досягти в найближчий рік, так як пізніше така система може бути неактуальною.

Планування змісту структури робіт. Основним інструментом для планування змісту структури робіт служить WBS діаграма – графічне подання згрупованих елементів проекту у вигляді пакета робіт, які ієрархічно пов'язані з

продуктом проекту. Побудуємо структуру WBS, у якій детально опишемо роботи, які потрібно виконати на кожному етапі створення проекту. Виконаємо декомпозицію робіт для даного проекту. Діаграма WBS зображена на рис. Б.1.

Планування структури організації, для впровадження готового проекту (OBS). Після побудови WBS розробимо організаційну структуру виконавців OBS. Організаційна структура проекту стосується тільки внутрішньої організаційної структури проекту і не стосується відносин проектних груп чи учасників з батьківськими організаціями. Діаграма OBS зображена на рис. Б.2. Список виконавців, що функціонують в проекті знаходиться в табл. Б.2.

Таблиця Б.2 – Матриця відповідності проекту

	OBS	Вакал С.М.	Лавров Є.А.	Захарченко Н.М.
WBS				
Визначення задачі проекту				
Визначення мети проекту				
Збір необхідної інформації про деталі проекту				
Аналіз зібраної інформації				
Проектування моделі				
Створення моделі системи				
Аналіз отриманої моделі та обговорення з замовником				
Правка та коректування моделі				
Визначення вимог				
Визначення інструментарію				
Планування WBS				
Планування OBS				
Календарний план				
Визначення бюджету				
Виникнення ризику				
Створення інформаційної системи				
Впровадження алгоритму роботи				
Написання коду				
Фікс знайдених проблем під час тестування				
Тестування інформаційної системи				
Оформлення ПЗ				
Інструкція користувача				
Впровадження підсистеми в експлуатацію				

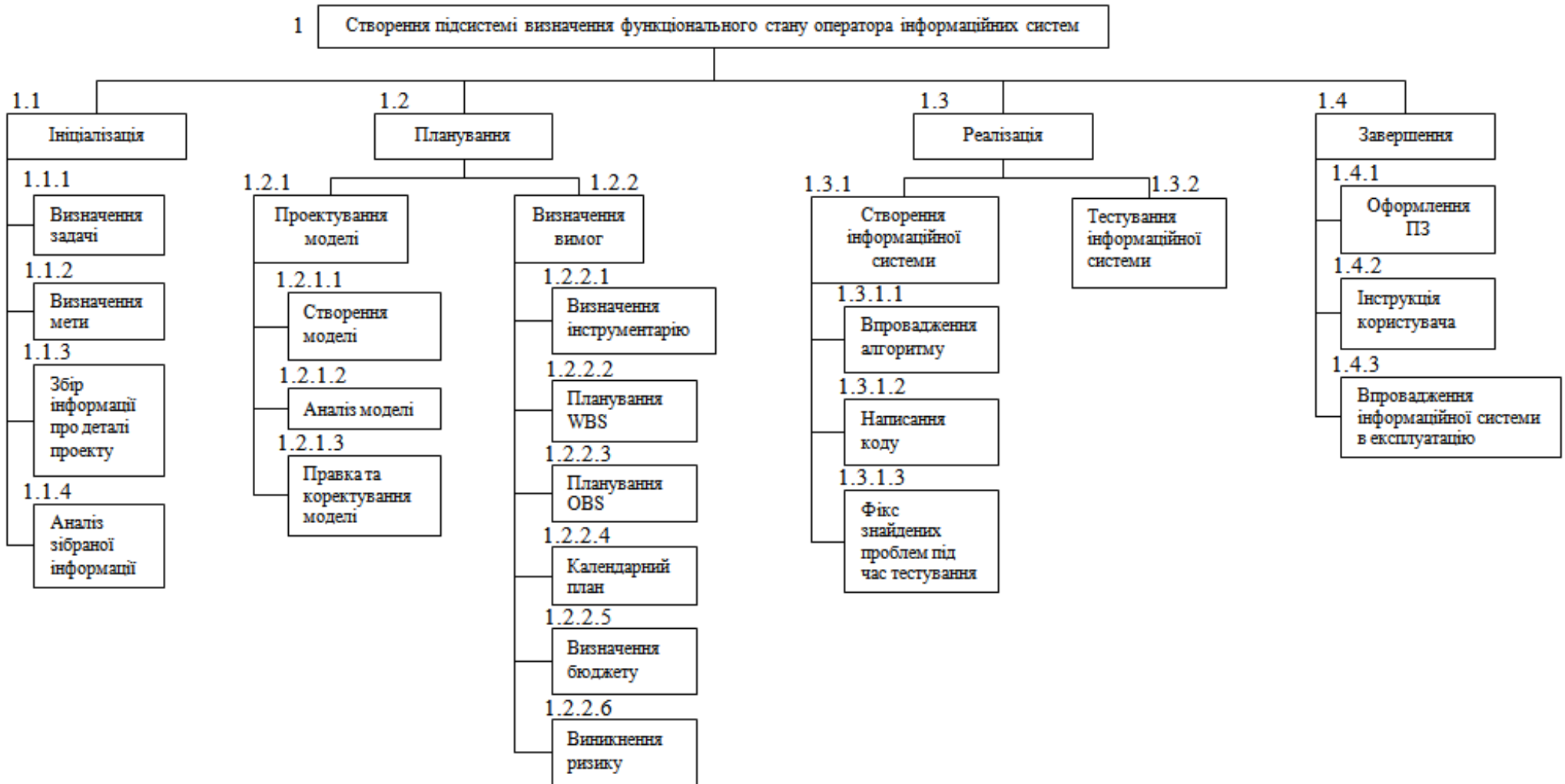


Рисунок Б.1 – WBS. Структура робіт проекту

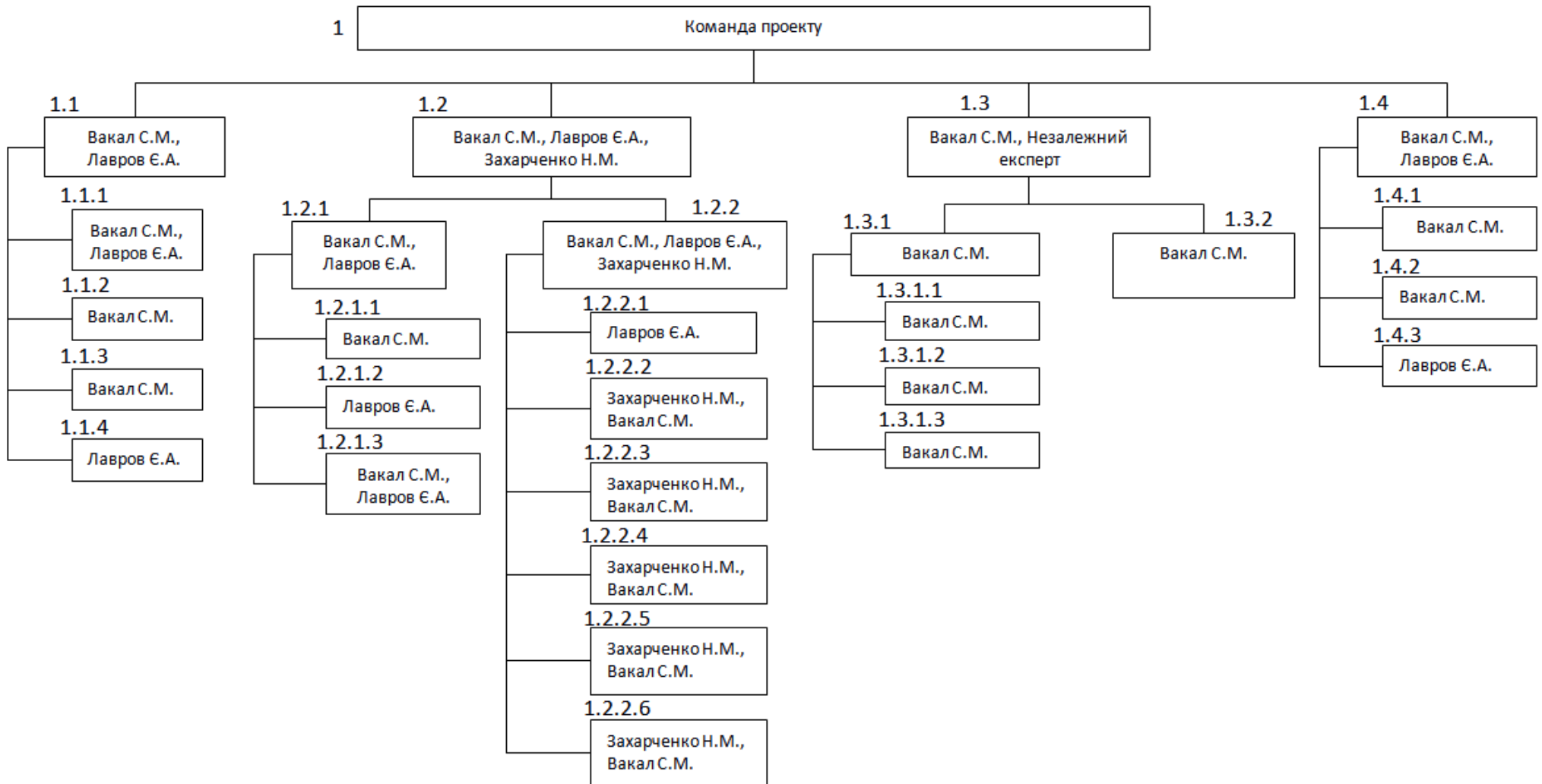


Рисунок Б.2 – Організаційна структура проекту (OBS)

Діаграма Ганта. Далі побудуємо календарний план виконання дипломного проекту. Найпоширеніший формат графіка в будь-якій галузі — діаграма Ганта. Цей графік дозволяє менеджерам проекту і всій команді розробників візуалізувати графіки часу і взаємозв'язок між окремими завданнями та етапами роботи над проектом. Тривалість виконання робіт зазначена в днях, але фактична тривалість виконання робіт приблизно дорівнює 2-3 години на день. Для того щоб мати реальне уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, з урахуванням вихідних та святкових днів, побудовано календарний графік. Діаграма Ганта та список робіт діаграми Ганта зображені на рис. Б.3-Б.5.

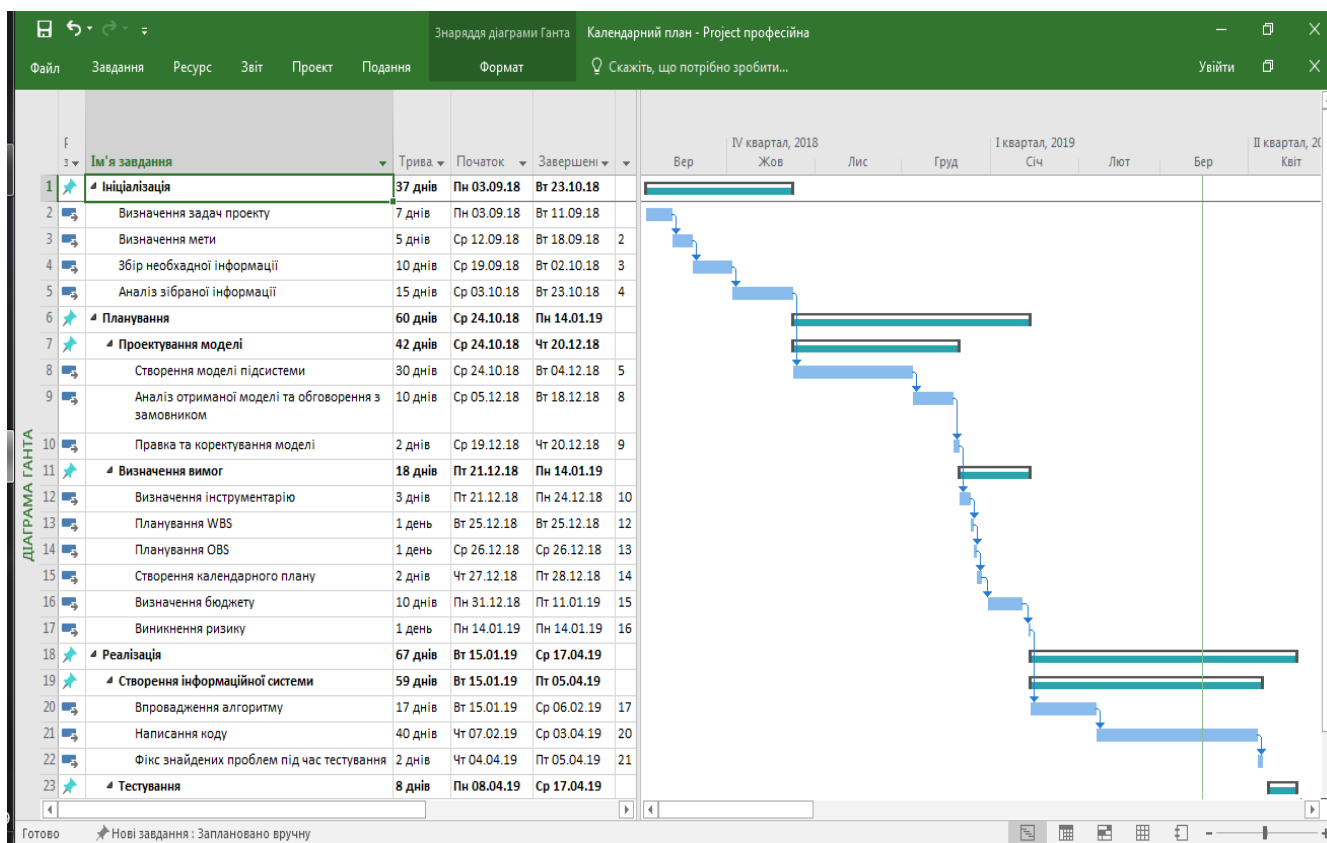


Рисунок Б.3 – Діаграма Ганта

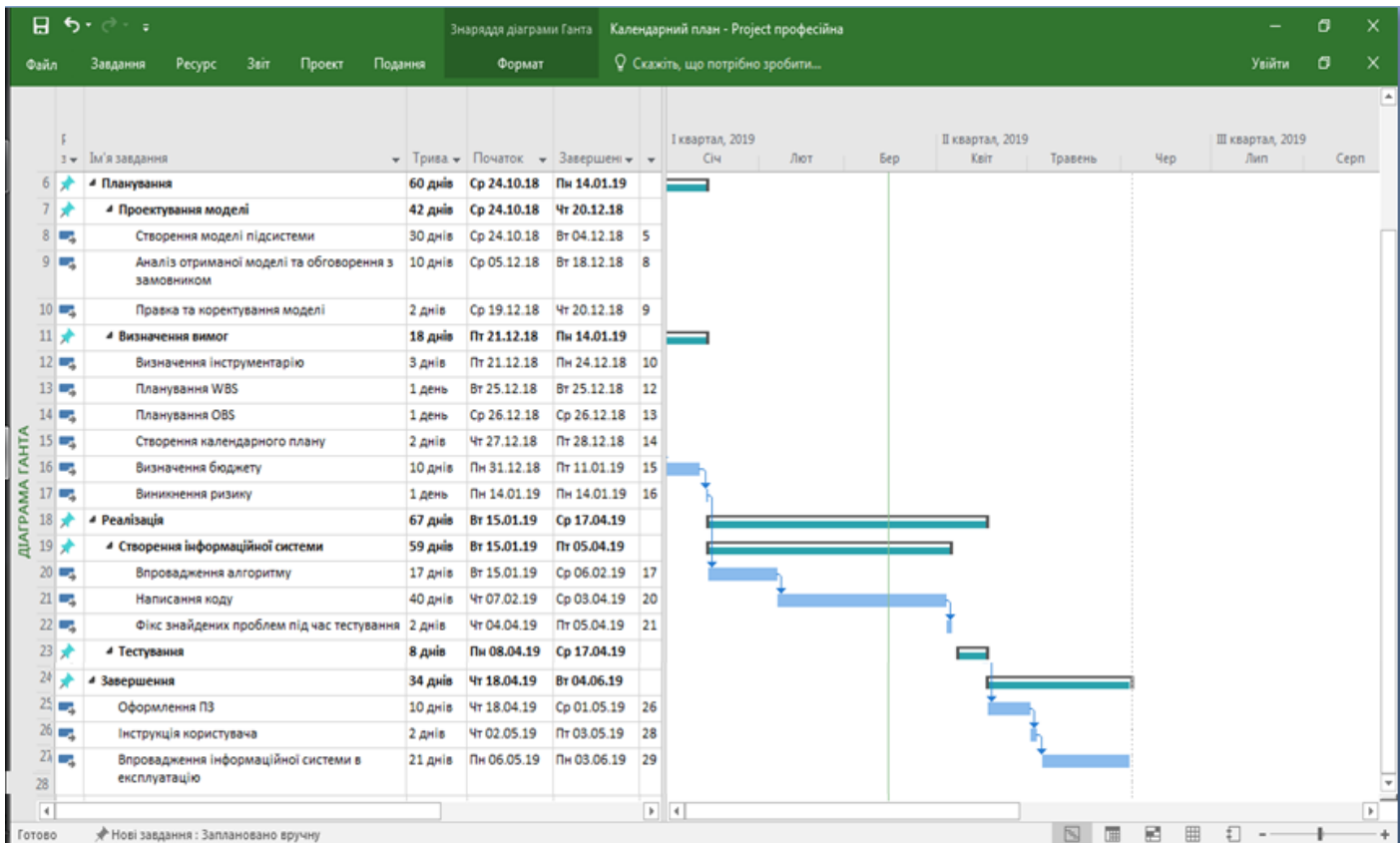


Рисунок Б.4 – Продовження діаграми Ганта

№	Ім'я завдання	Трива	Початок	Завершені
1	Ініціалізація	37 днів	Пн 03.09.18	Вт 23.10.18
2	Визначення задач проекту	7 днів	Пн 03.09.18	Вт 11.09.18
3	Визначення мети	5 днів	Ср 12.09.18	Вт 18.09.18
4	Збір необхідної інформації	10 днів	Ср 19.09.18	Вт 02.10.18
5	Аналіз зібраної інформації	15 днів	Ср 03.10.18	Вт 23.10.18
6	Планування	60 днів	Ср 24.10.18	Пн 14.01.19
7	Проектування моделі	42 днів	Ср 24.10.18	Чт 20.12.18
8	Створення моделі підсистеми	30 днів	Ср 24.10.18	Вт 04.12.18
9	Аналіз отриманої моделі та обговорення з замовником	10 днів	Ср 05.12.18	Вт 18.12.18
10	Правка та коректування моделі	2 днів	Ср 19.12.18	Чт 20.12.18
11	Визначення вимог	18 днів	Пт 21.12.18	Пн 14.01.19
12	Визначення інструментарію	3 днів	Пт 21.12.18	Пн 24.12.18
13	Планування WBS	1 день	Вт 25.12.18	Вт 25.12.18
14	Планування OBS	1 день	Ср 26.12.18	Ср 26.12.18
15	Створення календарного плану	2 днів	Чт 27.12.18	Пт 28.12.18
16	Визначення бюджету	10 днів	Пн 31.12.18	Пт 11.01.19
17	Виникнення ризику	1 день	Пн 14.01.19	Пн 14.01.19
18	Реалізація	67 днів	Вт 15.01.19	Ср 17.04.19
19	Створення інформаційної системи	59 днів	Вт 15.01.19	Пт 05.04.19
20	Впровадження алгоритму	17 днів	Вт 15.01.19	Ср 06.02.19
21	Написання коду	40 днів	Чт 07.02.19	Ср 03.04.19
22	Фікс знайдених проблем під час тестування	2 днів	Чт 04.04.19	Пт 05.04.19
23	Тестування	8 днів	Пн 08.04.19	Ср 17.04.19
27	Завершення	34 днів	Чт 18.04.19	Вт 04.06.19
28	Оформлення ПЗ	10 днів	Чт 18.04.19	Ср 01.05.19
29	Інструкція користувача	2 днів	Чт 02.05.19	Пт 03.05.19
30	Впровадження інформаційної системи в експлуатацію	21 днів	Пн 06.05.19	Пн 03.06.19

Рисунок Б.5 – Список робіт для побудови діаграми Ганта

Аналіз ризиків. Однією з основних завдань, які виконують в рамках управління IT-проектами, є управління ризиками проектної діяльності, або управління ризиками проекту. Це завдання не відділяється від інших функцій управління IT-проектами. При визначенні фінансових потреб, обчислення кошторису і бюджету, підготовка та укладення контрактів, при контролі за реалізацією проекту стоїть завдання захисту учасників проектної діяльності від різних видів ризиків. Саме тому проблеми дослідження та управління ризиками в проектній діяльності є важливими і актуальними як з точки зору теорії, так і щодо практичних застосувань.

Ризик проекту - це невизначена подія або умова, настання якого негативно чи позитивно позначається на цілях проекту, таких як зміст, розклад, вартість і якість. Ризик може бути викликаний однією або декількома причинами і в разі виникнення може вплинути на один або декілька аспектів. Причиною може бути існуюча або потенційна вимога, допущення, обмеження або умова, яку створює ймовірність негативних чи позитивних наслідків. Наприклад, причиною ризику може бути необхідність отримання дозвільної документації в галузі охорони навколишнього середовища або недолік персоналу, залученого для розробки проекту. Ризиком в першому випадку буде затримка з видачею дозволу контролюючим органом, а по-друге, в разі сприятливої можливості, додатковий персонал, який може бути притягнутий до розробки проекту, може стати доступним для призначення на проект. Виникнення будь-якого з цих точно не відомих заздалегідь подій може вплинути на проект, його зміст, вартість, розклад, якість чи виконання. До умов виникнення ризиків можуть також ставитися аспекти середовища організації або проекту, що сприяють збільшенню ризику.

Перший етап в управлінні ризиками є їх ідентифікація (табл. Б.3). В результаті аналізу були виявлені ризики, які впливають на виконання проекту, їх причини виникнення та можливі вирішення.

Таблиця Б.3 – Ідентифікація ризиків

№ ризику	Назва ризику	Причини виникнення	Шляхи вирішення
1	Вихід з ладу апаратного забезпечення	Вихід із ладу компонентів ПК/ноутбука	Звернення до сервісного центру для ремонту апаратного забезпечення
2	Видалення програми/звіту	1. Людський фактор 2. Зараження вірусами	1. Використовувати засоби для резервного копіювання (зберегти копії на флеш-носії, також збереження копії, наприклад, на Google Диск 2. Своєчасне оновлення/інсталяція антивірусного забезпечення
3	Збій у роботі програми	Некоректне написання коду	Протестувати програму на помилки та виправити їх
4	Хвороба студента-виконавця	1. Захворювання 2. Нещасний випадок	Забезпечити своєчасне звернення до лікаря
5	Поїздка у відрядження студента/керівника проекту	Термінове відрядження до іншого міста/країни	

Другий етап передбачає розрахунок кількісної оцінки виявлених ризиків, що може бути виражена у відносному або абсолютному рівні витрат та вимірюється ймовірністю виникнення та ступеню впливу ризику при його появі.

Імовірність виникнення ризику буває:

- Слабо ймовірна
- Малоймовірна
- Імовірна
- Більш ймовірна
- Майже можливе

Величина втрат має такі рівні

- Мінімальна
- Низька
- Середня
- Висока
- Максимальна

Ступінь впливу визначається як добуток імовірності виникнення та величини втрат (табл.Б.4 – Б.5).

Таблиця Б.4 – Матриця «Імовірність – втрати»

Імовірність	5	5	10	15	20	25
	4	4	8	12	16	20
	3	3	6	9	12	15
	2	2	4	6	8	10
	1	1	2	3	4	5
Втрати		1	2	3	4	5

Прийнятні – виділені зеленим кольором (1-4).

Виправдати – виділені помаранчевим кольором (5-10).

Неприпустимі – виділені червоним кольором (12-25).

Таблиця Б.5 – Класифікація ризиків по величині втрат

Види ризиків	I_g (бали)	Величина втрат (% від планової прибутковості втрат)
Мінімальна	1	(0;10]
Низька	2	(10;40]
Середня	3	(40;60]
Висока	4	(60;90]
Максимальна	5	(90;100]

Оцінка ризику розраховується за формулою:

$$R = P_g \cdot I_g, \quad (\text{Б.1})$$

де R – індекс ризику (у балах);

P_g – імовірність виникнення ризику згідно класифікації (бали);

I_g – величина втрат у відповідності до класифікації ризиків.

1. Вихід з ладу апаратного забезпечення: $R = 25 \cdot 0,5 = 12,5$.
2. Видалення програми/звіту: $R = 25 \cdot 0,4 = 10$.
3. Збій у роботі програми: $R = 25 \cdot 0,5 = 12,5$.
4. Хвороба студента-виконавця: $R = 20 \cdot 0,2 = 4$.
5. Поїздка у відрядження студента/керівника: $R = 15 \cdot 0,4 = 6$.

Із розрахунків випливає, що найбільш небезпечними є ризики, які пов'язані із виходом з ладу робочого ПК чи ноутбука виконавця робіт, збій у роботі програм, та видалення програми чи звіту з пам'яті комп'ютера. Тому основними заходами щодо передбаченню цих небезпек необхідно виконати наступні заходи:

1. Використовувати засоби для резервного копіювання (зберегти копії на флеш-носії, також збереження копії, наприклад, на Google Диск).
2. Своєчасне оновлення/інсталяція антивірусного забезпечення.
3. Тестувати програму на наявність помилок, у разі знаходження виправляти їх.
4. Звернення до сервісного центру для ремонту апаратного забезпечення.

Захворювання виконавця роботи також можна проігнорувати, тому що нещасний випадок малоімовірне, а використання ноутбука дозволяє працювати автономно навіть у лікарні.

Поїздка у відрядження студента/керівника проекту може призвести до затримки написання/здачі проекту або перевірки результатів роботи студента. Даний ризик також може бути критичним, але малоімовірним. Саме тому ним можна знехтувати.

ДОДАТОК В. Лістинг програмного коду

Код для головного вікна програми «Визначення ФС» (Form3)

```
Imports System
Imports System.IO
Imports System.Windows.Forms.ComboBox
Module Module1
    Public Direct As String
End Module

Public Class Form3

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        If (Not System.IO.Directory.Exists("C:\FS")) Then
            System.IO.Directory.CreateDirectory("C:\FS")
            MsgBox("Папку C:\FS створено!", MessageBoxIcon.Information)
        End If
        For Each dir_name As String In Directory.GetDirectories("C:\FS\")
            ComboBox1.Items.Add(dir_name)
            ComboBox2.Items.Add(dir_name)
        Next
    End Sub

    Private Sub ComboBox1_SelectedValueChanged(ByVal sender As Object, ByVal e As System.EventArgs)
Handles ComboBox1.SelectedValueChanged
        Direct = ComboBox1.Text
        Form4.Visible = True
        Me.Visible = False
    End Sub

    Private Sub someButton_MouseHover(ByVal sender As Object, ByVal e As System.EventArgs) _
Handles Button3.MouseHover
        Dim toolTip As New ToolTip()
        toolTip.SetToolTip(Button3, "Створити папку")
        toolTip.SetToolTip(Button4, "Видалити папку")
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        If TextBox1.Text = "" Then
            MsgBox("Ви не вказали назву створюваної папки!", MessageBoxIcon.Warning)
        ElseIf (Not System.IO.Directory.Exists("C:\FS\" & TextBox1.Text)) Then
            System.IO.Directory.CreateDirectory("C:\FS\" & TextBox1.Text)
            MsgBox("Папку збережено!", MessageBoxIcon.Information)
            ComboBox1.Items.Clear()
            ComboBox2.Items.Clear()
            For Each dir_name As String In Directory.GetDirectories("C:\FS\")
                ComboBox1.Items.Add(dir_name)
                ComboBox2.Items.Add(dir_name)
            Next
            TextBox1.Text = ""
        ElseIf MsgBox("Вибачте, але папка з такою назвою вже існує!", MessageBoxIcon.Warning) Then
        End If

    End Sub

End Sub
```

```

Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As KeyPressEventArgs) Handles
TextBox1.KeyPress
    If Not Char.IsLetter(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then
        e.KeyChar = ""
    End If
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
    Panel1.Visible = True
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
    Panel2.Visible = True
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
    If ComboBox2.SelectedIndex = -1 Then
        MsgBox("Оберіть папку з випадного списку!", MessageBoxIcon.Warning)
    Else : CreateObject("Scripting.FileSystemObject").GetFolder(ComboBox2.Text.ToString()).Delete()
        ComboBox1.Items.Clear()
        ComboBox2.Items.Clear()
        For Each dir_name As String In Directory.GetDirectories("C:\FS\")
            ComboBox1.Items.Add(dir_name)
            ComboBox2.Items.Add(dir_name)
        Next
        ComboBox2.Text = ""
        MsgBox("Папку " & ComboBox2.Text.ToString() & " видалено!", MessageBoxIcon.Information)
    End If
End Sub

Private Sub ПроАвтораToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ПроАвтораToolStripMenuItem.Click
    Form6.Visible = True
    Me.Visible = False
End Sub

End Class

```

Код для проміжного вікна (Form4)

```

Imports System.IO

Public Class Form4

    Private Sub Form4_Closing(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.FormClosing
        Form3.Visible = True
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        Form1.Visible = True
        Me.Visible = False
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click

```

```

If Directory.Exists(Direct) Then
    If Directory.Exists(Direct & "\Навчання НМ") Then
        If Not File.Exists(Direct & "\Навчання НМ\адекватний_train.txt") Or Not File.Exists(Direct &
"\Навчання НМ\бібліотека_train.txt") Or Not File.Exists(Direct & "\Навчання НМ\відеопотік_train.txt") Or Not
File.Exists(Direct & "\Навчання НМ\гермомішок_train.txt") Or Not File.Exists(Direct & "\Навчання
НМ\дендропарк_train.txt") Or Not File.Exists(Direct & "\Навчання НМ\зубочистка_train.txt") Or Not File.Exists(Direct &
"\Навчання НМ\катастрофа_train.txt") Or Not File.Exists(Direct & "\Навчання НМ\латифундія_train.txt") Or Not
File.Exists(Direct & "\Навчання НМ\менеджмент_train.txt") Or Not File.Exists(Direct & "\Навчання
НМ\наповнення_train.txt") Then
            MessageBox.Show("Перевірте, чи всі файли заповнені!", "Помилка!", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
        Else
            Dim m1String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\адекватний_train.txt",
System.Text.Encoding.Default)
            Dim b1 = m1String.Length
            Dim m2String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\бібліотека_train.txt",
System.Text.Encoding.Default)
            Dim b2 = m2String.Length
            Dim m3String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\відеопотік_train.txt",
System.Text.Encoding.Default)
            Dim b3 = m3String.Length
            Dim m4String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\гермомішок_train.txt",
System.Text.Encoding.Default)
            Dim b4 = m4String.Length
            Dim m5String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\дендропарк_train.txt",
System.Text.Encoding.Default)
            Dim b5 = m5String.Length
            Dim m6String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\зубочистка_train.txt",
System.Text.Encoding.Default)
            Dim b6 = m6String.Length
            Dim m7String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\катастрофа_train.txt",
System.Text.Encoding.Default)
            Dim b7 = m7String.Length
            Dim m8String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\латифундія_train.txt",
System.Text.Encoding.Default)
            Dim b8 = m8String.Length
            Dim m9String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\менеджмент_train.txt",
System.Text.Encoding.Default)
            Dim b9 = m9String.Length
            Dim m10String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\наповнення_train.txt",
System.Text.Encoding.Default)
            Dim b10 = m10String.Length
            If b1 = 0 Or b2 = 0 Or b3 = 0 Or b4 = 0 Or b5 = 0 Or b6 = 0 Or b7 = 0 Or b8 = 0 Or b9 = 0 Or b10 = 0
Then
                MessageBox.Show("Перевірте, чи всі файли заповнені!", "Помилка!", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
            Else
                Form2.Visible = True
                Me.Visible = False
            End If
        End If
        ElseIf MsgBox("Вибачте, папки " & Direct & "\Навчання НМ немає. Виконайте навчання Нейронної
мережі!", MessageBoxIcon.Warning) Then
            End If
        ElseIf MsgBox("Вибачте, папка " & Direct & " пуста. Виконайте навчання Нейронної мережі!",
MessageBoxIcon.Warning) Then
            End If
        End Sub
    End Class

```

Код для «Персональне налаштування системи» (Form1)

```
Imports System.IO
Imports System.Runtime.InteropServices
```

```
Public Class Form1
```

```
Dim kd1, kd2, kd3, kd4, kd5, kd6, kd7, kd8, kd9, kd10 As Date 'событие нажатия на кнопку
Dim ku1, ku2, ku3, ku4, ku5, ku6, ku7, ku8, ku9, ku10 As Date 'событие отпуща кнопки
Dim tn1, tn2, tn3, tn4, tn5, tn6, tn7, tn8, tn9, tn10 As Single 'время между нажатиями кнопок
Dim tu1, tu2, tu3, tu4, tu5, tu6, tu7, tu8, tu9, tu10 As Single 'время удержания кнопок
Dim reader As StreamReader
```

```
Private Sub TextBox1_KeyDown(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
```

```
    If e.KeyCode = 70 Then
        kd1 = Now()
```

```
    End If
```

```
    If e.KeyCode = 76 Then
        kd2 = Now()
```

```
    End If
```

```
    If e.KeyCode = 84 Then
        kd3 = Now()
```

```
    End If
```

```
    If e.KeyCode = 82 Then
        kd4 = Now()
```

```
    End If
```

```
    If e.KeyCode = 68 Then
        kd5 = Now()
```

```
    End If
```

```
    If e.KeyCode = 70 Then
        kd6 = Now()
```

```
    End If
```

```
    If e.KeyCode = 78 Then
        kd7 = Now()
```

```
    End If
```

```
    If e.KeyCode = 89 Then
        kd8 = Now()
```

```
    End If
```

```
    If e.KeyCode = 66 Then
        kd9 = Now()
```

```
    End If
```

```
    If e.KeyCode = 81 Then
        kd10 = Now()
```

```
    End If
```

```
End Sub
```

```
Private Sub TextBox10_KeyUp(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox10.KeyUp
```

```
    If e.KeyCode = 89 Then
        ku1 = Now()
```

```
    End If
```

```
    If e.KeyCode = 70 Then
        ku2 = Now()
```

```
    End If
```

```
    If e.KeyCode = 71 Then
        ku3 = Now()
```

```
    End If
```

```
    If e.KeyCode = 74 Then
        ku4 = Now()
```

```
    End If
```

```
    If e.KeyCode = 68 Then
        ku5 = Now()
```

```
    End If
```

```

If e.KeyCode = 89 Then
    ku6 = Now()
End If
If e.KeyCode = 84 Then
    ku7 = Now()
End If
If e.KeyCode = 89 Then
    ku8 = Now()
End If
If e.KeyCode = 89 Then
    ku9 = Now()
End If
If e.KeyCode = 90 Then
    ku10 = Now()
End If
End Sub

```

```

Private Sub Form1_Closed(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.FormClosed
    Form4.Visible = True
End Sub

```

```

Private Sub someButton_MouseHover(ByVal sender As Object, ByVal e As System.EventArgs) _
Handles Button1.MouseHover
    Dim toolTip As New ToolTip()
    toolTip.SetToolTip(Button1, "Додати в файл")
    toolTip.SetToolTip(Button2, "Визначити поточний функціональний стан")
    toolTip.SetToolTip(Button3, "На головну")
    toolTip.SetToolTip(Button4, "Додати в файл")
    toolTip.SetToolTip(Button5, "Додати в файл")
    toolTip.SetToolTip(Button6, "Додати в файл")
    toolTip.SetToolTip(Button7, "Додати в файл")
    toolTip.SetToolTip(Button8, "Додати в файл")
    toolTip.SetToolTip(Button9, "Додати в файл")
    toolTip.SetToolTip(Button10, "Додати в файл")
    toolTip.SetToolTip(Button11, "Додати в файл")
    toolTip.SetToolTip(Button12, "Додати в файл")
    toolTip.SetToolTip(Button13, "Видалити файли")
End Sub

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
    If Not System.IO.Directory.Exists(Direct & "\Навчання НМ") Then
        System.IO.Directory.CreateDirectory(Direct & "\Навчання НМ")
    ElseIf (Not File.Exists(Direct & "\Навчання НМ\" & Label1.Text & "_train.txt")) Then
        Dim g
        g = File.Create(Direct & "\Навчання НМ\" & Label1.Text & "_train.txt")
        g.Close()
    End If
    Try
        Dim m1String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label1.Text & "_train.txt",
System.Text.Encoding.Default)
        Dim b1 = m1String.Length
        If TextBox1.Text = Label1.Text Then
            tn1 = 0
            tu1 = (ku1 - kd1).Milliseconds / 1000
            tn2 = (kd2 - ku1).Milliseconds / 1000
            tu2 = (ku2 - kd2).Milliseconds / 1000
            tn3 = (kd3 - ku2).Milliseconds / 1000
            tu3 = (ku3 - kd3).Milliseconds / 1000
            tn4 = (kd4 - ku3).Milliseconds / 1000
            tu4 = (ku4 - kd4).Milliseconds / 1000
            tn5 = (kd5 - ku4).Milliseconds / 1000

```

```

tu5 = (ku5 - kd5).Milliseconds / 1000
tn6 = (kd6 - ku5).Milliseconds / 1000
tu6 = (ku6 - kd6).Milliseconds / 1000
tn7 = (kd7 - ku6).Milliseconds / 1000
tu7 = (ku7 - kd7).Milliseconds / 1000
tn8 = (kd8 - ku7).Milliseconds / 1000
tu8 = (ku8 - kd8).Milliseconds / 1000
tn9 = (kd9 - ku8).Milliseconds / 1000
tu9 = (ku9 - kd9).Milliseconds / 1000
tn10 = (kd10 - ku9).Milliseconds / 1000
tu10 = (ku10 - kd10).Milliseconds / 1000
Dim writer As New StreamWriter(Direct & "\Навчання НМ\" & Label1.Text & "_train.txt", True,
System.Text.Encoding.GetEncoding(1251))
writer.WriteLine(tn1 & vbTab & tu1 & vbTab & tn2 & vbTab & tu2 & vbTab & tn3 & vbTab & tu3 &
vbTab & tn4 & vbTab & tu4 & vbTab & tn5 & vbTab & tu5 & vbTab & tn6 & vbTab & tu6 & vbTab & tn7 & vbTab & tu7 &
vbTab & tn8 & vbTab & tu8 & vbTab & tn9 & vbTab & tu9 & vbTab & tn10 & vbTab & tu10) ' Записуємо в файл текст
writer.Close() ' Закриваємо файл
Dim Txt As String = IO.File.ReadAllText(Direct & "\Навчання НМ\" & Label1.Text & "_train.txt",
System.Text.Encoding.Default)
Txt = Txt.Replace(", ", ".")
IO.File.WriteAllText(Direct & "\Навчання НМ\" & Label1.Text & "_train.txt", Txt,
System.Text.Encoding.Default)

Label12.Text = "Кількість введень = " & b1 + 1
TextBox1.Text = ""
TextBox1.Select()
Else : MessageBox.Show("Будь ласка, перевірте правильність введення фрази для навчання НМ",
"Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
End If
Catch ex As Exception
If ex.TargetSite.Name = "WinIOError" Then MsgBox("Файл зайнятий іншим додатком!",
MessageBoxIcon.Information)
End Try
End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
If Not File.Exists(Direct & "\Навчання НМ\" & Label1.Text & "_train.txt") Then
Label12.Text = "Кількість введень = 0"
Else : Dim m1String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label1.Text & "_train.txt",
System.Text.Encoding.Default)
Dim b1 = m1String.Length
Label12.Text = "Кількість введень = " & b1
End If
If Not File.Exists(Direct & "\Навчання НМ\" & Label2.Text & "_train.txt") Then
Label13.Text = "Кількість введень = 0"
Else : Dim m2String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label2.Text & "_train.txt",
System.Text.Encoding.Default)
Dim b2 = m2String.Length
Label13.Text = "Кількість введень = " & b2
End If
If Not File.Exists(Direct & "\Навчання НМ\" & Label3.Text & "_train.txt") Then
Label14.Text = "Кількість введень = 0"
Else : Dim m3String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label3.Text & "_train.txt",
System.Text.Encoding.Default)
Dim b3 = m3String.Length
Label14.Text = "Кількість введень = " & b3
End If
If Not File.Exists(Direct & "\Навчання НМ\" & Label4.Text & "_train.txt") Then
Label15.Text = "Кількість введень = 0"
Else : Dim m4String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label4.Text & "_train.txt",
System.Text.Encoding.Default)
Dim b4 = m4String.Length
Label15.Text = "Кількість введень = " & b4

```

```

End If
If Not File.Exists(Direct & "\Навчання НМ\" & Label5.Text & "_train.txt") Then
    Label16.Text = "Кількість введень = 0"
Else : Dim m5String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label5.Text & "_train.txt",
System.Text.Encoding.Default)
    Dim b5 = m5String.Length
    Label16.Text = "Кількість введень = " & b5
End If
If Not File.Exists(Direct & "\Навчання НМ\" & Label6.Text & "_train.txt") Then
    Label17.Text = "Кількість введень = 0"
Else : Dim m6String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label6.Text & "_train.txt",
System.Text.Encoding.Default)
    Dim b6 = m6String.Length
    Label17.Text = "Кількість введень = " & b6
End If
If Not File.Exists(Direct & "\Навчання НМ\" & Label7.Text & "_train.txt") Then
    Label18.Text = "Кількість введень = 0"
Else : Dim m7String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label7.Text & "_train.txt",
System.Text.Encoding.Default)
    Dim b7 = m7String.Length
    Label18.Text = "Кількість введень = " & b7
End If
If Not File.Exists(Direct & "\Навчання НМ\" & Label8.Text & "_train.txt") Then
    Label19.Text = "Кількість введень = 0"
Else : Dim m8String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label8.Text & "_train.txt",
System.Text.Encoding.Default)
    Dim b8 = m8String.Length
    Label19.Text = "Кількість введень = " & b8
End If
If Not File.Exists(Direct & "\Навчання НМ\" & Label9.Text & "_train.txt") Then
    Label20.Text = "Кількість введень = 0"
Else : Dim m9String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label9.Text & "_train.txt",
System.Text.Encoding.Default)
    Dim b9 = m9String.Length
    Label20.Text = "Кількість введень = " & b9
End If
If Not File.Exists(Direct & "\Навчання НМ\" & Label10.Text & "_train.txt") Then
    Label21.Text = "Кількість введень = 0"
Else : Dim m10String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label10.Text & "_train.txt",
System.Text.Encoding.Default)
    Dim b10 = m10String.Length
    Label21.Text = "Кількість введень = " & b10
End If
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click ' FS
    If Directory.Exists(Direct) Then
        If Directory.Exists(Direct & "\Навчання НМ") Then
            If Not File.Exists(Direct & "\Навчання НМ\адекватний_train.txt") Or Not File.Exists(Direct &
"\Навчання НМ\бібліотека_train.txt") Or Not File.Exists(Direct & "\Навчання НМ\відеопотік_train.txt") Or Not
File.Exists(Direct & "\Навчання НМ\гермомішок_train.txt") Or Not File.Exists(Direct & "\Навчання
НМ\дендропарк_train.txt") Or Not File.Exists(Direct & "\Навчання НМ\зубочистка_train.txt") Or Not File.Exists(Direct &
"\Навчання НМ\катастрофа_train.txt") Or Not File.Exists(Direct & "\Навчання НМ\латифундія_train.txt") Or Not
File.Exists(Direct & "\Навчання НМ\менеджмент_train.txt") Or Not File.Exists(Direct & "\Навчання
НМ\наповнення_train.txt") Then
                MessageBox.Show("Перевірте, чи всі файли заповнені!", "Помилка!", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
            Else
                Dim m1String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\адекватний_train.txt",
System.Text.Encoding.Default)
                Dim b1 = m1String.Length

```

```

        Dim m2String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\бібліотека_train.txt",
System.Text.Encoding.Default)
        Dim b2 = m2String.Length
        Dim m3String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\відеопотік_train.txt",
System.Text.Encoding.Default)
        Dim b3 = m3String.Length
        Dim m4String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\гермомішок_train.txt",
System.Text.Encoding.Default)
        Dim b4 = m4String.Length
        Dim m5String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\дендропарк_train.txt",
System.Text.Encoding.Default)
        Dim b5 = m5String.Length
        Dim m6String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\зубочистка_train.txt",
System.Text.Encoding.Default)
        Dim b6 = m6String.Length
        Dim m7String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\катастрофа_train.txt",
System.Text.Encoding.Default)
        Dim b7 = m7String.Length
        Dim m8String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\латифундія_train.txt",
System.Text.Encoding.Default)
        Dim b8 = m8String.Length
        Dim m9String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\менеджмент_train.txt",
System.Text.Encoding.Default)
        Dim b9 = m9String.Length
        Dim m10String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\наповнення_train.txt",
System.Text.Encoding.Default)
        Dim b10 = m10String.Length
        If b1 = 0 Or b2 = 0 Or b3 = 0 Or b4 = 0 Or b5 = 0 Or b6 = 0 Or b7 = 0 Or b8 = 0 Or b9 = 0 Or b10 = 0
Then
            MessageBox.Show("Перевірте, чи всі файли заповнені!", "Помилка!", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
        Else
            Form2.Visible = True
            Me.Visible = False
        End If
    End If
    ElseIf MessageBox.Show("Вибачте, папки " & Direct & "\Навчання НМ немає. Виконайте навчання
Нейронної мережі!", "Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Warning) Then
    End If
    ElseIf MessageBox.Show("Вибачте, папка " & Direct & " пуста. Виконайте навчання Нейронної мережі!",
"Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Warning) Then
    End If
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
    Form3.Visible = True
    Me.Visible = False
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
    If Not System.IO.Directory.Exists(Direct & "\Навчання НМ") Then
        System.IO.Directory.CreateDirectory(Direct & "\Навчання НМ")
    ElseIf (Not File.Exists(Direct & "\Навчання НМ\" & Label2.Text & "_train.txt")) Then
        Dim g
        g = File.Create(Direct & "\Навчання НМ\" & Label2.Text & "_train.txt")
        g.Close()
    End If
    Try
        Dim m2String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label2.Text & "_train.txt",
System.Text.Encoding.Default)
        Dim b2 = m2String.Length

```



```

If TextBox2.Text = Label2.Text Then
    tn1 = 0
    tu1 = (ku1 - kd1).Milliseconds / 1000
    tn2 = (kd2 - ku1).Milliseconds / 1000
    tu2 = (ku2 - kd2).Milliseconds / 1000
    tn3 = (kd3 - ku2).Milliseconds / 1000
    tu3 = (ku3 - kd3).Milliseconds / 1000
    tn4 = (kd4 - ku3).Milliseconds / 1000
    tu4 = (ku4 - kd4).Milliseconds / 1000
    tn5 = (kd5 - ku4).Milliseconds / 1000
    tu5 = (ku5 - kd5).Milliseconds / 1000
    tn6 = (kd6 - ku5).Milliseconds / 1000
    tu6 = (ku6 - kd6).Milliseconds / 1000
    tn7 = (kd7 - ku6).Milliseconds / 1000
    tu7 = (ku7 - kd7).Milliseconds / 1000
    tn8 = (kd8 - ku7).Milliseconds / 1000
    tu8 = (ku8 - kd8).Milliseconds / 1000
    tn9 = (kd9 - ku8).Milliseconds / 1000
    tu9 = (ku9 - kd9).Milliseconds / 1000
    tn10 = (kd10 - ku9).Milliseconds / 1000
    tu10 = (ku10 - kd10).Milliseconds / 1000
    Dim writer As New StreamWriter(Direct & "\Навчання НМ\" & Label2.Text & "_train.txt", True,
System.Text.Encoding.GetEncoding(1251))
    writer.WriteLine(tn1 & vbTab & tu1 & vbTab & tn2 & vbTab & tu2 & vbTab & tn3 & vbTab & tu3 &
vbTab & tn4 & vbTab & tu4 & vbTab & tn5 & vbTab & tu5 & vbTab & tn6 & vbTab & tu6 & vbTab & tn7 & vbTab & tu7 &
vbTab & tn8 & vbTab & tu8 & vbTab & tn9 & vbTab & tu9 & vbTab & tn10 & vbTab & tu10) 'Записуємо в файл текст
    writer.Close() 'Закриваємо файл
    Dim Txt As String = IO.File.ReadAllText(Direct & "\Навчання НМ\" & Label2.Text & "_train.txt",
System.Text.Encoding.Default)
    Txt = Txt.Replace(", ", ".")
    IO.File.WriteAllText(Direct & "\Навчання НМ\" & Label2.Text & "_train.txt", Txt,
System.Text.Encoding.Default)

    Label13.Text = "Кількість введень = " & b2 + 1
    TextBox2.Text = ""
    TextBox2.Select()
Else : MessageBox.Show("Будь ласка, перевірте правильність введення фрази для навчання НМ",
"Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
End If
Catch ex As Exception
If ex.TargetSite.Name = "WinIOError" Then MsgBox("Файл зайнятий іншим додатком!",
MessageBoxIcon.Information)
End Try
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button5.Click
If Not System.IO.Directory.Exists(Direct & "\Навчання НМ") Then
System.IO.Directory.CreateDirectory(Direct & "\Навчання НМ")
Elseif (Not File.Exists(Direct & "\Навчання НМ\" & Label3.Text & "_train.txt")) Then
Dim g
g = File.Create(Direct & "\Навчання НМ\" & Label3.Text & "_train.txt")
g.Close()
End If
Try
Dim m3String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label3.Text & "_train.txt",
System.Text.Encoding.Default)
Dim b3 = m3String.Length
If TextBox3.Text = Label3.Text Then
tn1 = 0
tu1 = (ku1 - kd1).Milliseconds / 1000
tn2 = (kd2 - ku1).Milliseconds / 1000
tu2 = (ku2 - kd2).Milliseconds / 1000

```

```

tn3 = (kd3 - ku2).Milliseconds / 1000
tu3 = (ku3 - kd3).Milliseconds / 1000
tn4 = (kd4 - ku3).Milliseconds / 1000
tu4 = (ku4 - kd4).Milliseconds / 1000
tn5 = (kd5 - ku4).Milliseconds / 1000
tu5 = (ku5 - kd5).Milliseconds / 1000
tn6 = (kd6 - ku5).Milliseconds / 1000
tu6 = (ku6 - kd6).Milliseconds / 1000
tn7 = (kd7 - ku6).Milliseconds / 1000
tu7 = (ku7 - kd7).Milliseconds / 1000
tn8 = (kd8 - ku7).Milliseconds / 1000
tu8 = (ku8 - kd8).Milliseconds / 1000
tn9 = (kd9 - ku8).Milliseconds / 1000
tu9 = (ku9 - kd9).Milliseconds / 1000
tn10 = (kd10 - ku9).Milliseconds / 1000
tu10 = (ku10 - kd10).Milliseconds / 1000
Dim writer As New StreamWriter(Direct & "\Навчання НМ\" & Label3.Text & "_train.txt", True,
System.Text.Encoding.GetEncoding(1251))
writer.WriteLine(tn1 & vbTab & tu1 & vbTab & tn2 & vbTab & tu2 & vbTab & tn3 & vbTab & tu3 &
vbTab & tn4 & vbTab & tu4 & vbTab & tn5 & vbTab & tu5 & vbTab & tn6 & vbTab & tu6 & vbTab & tn7 & vbTab & tu7 &
vbTab & tn8 & vbTab & tu8 & vbTab & tn9 & vbTab & tu9 & vbTab & tn10 & vbTab & tu10) ' Записуємо в файл текст
writer.Close() ' Закриваємо файл
Dim Txt As String = IO.File.ReadAllText(Direct & "\Навчання НМ\" & Label3.Text & "_train.txt",
System.Text.Encoding.Default)
Txt = Txt.Replace(", ", ".")
IO.File.WriteAllText(Direct & "\Навчання НМ\" & Label3.Text & "_train.txt", Txt,
System.Text.Encoding.Default)

Label14.Text = "Кількість введень = " & b3 + 1
TextBox3.Text = ""
TextBox3.Select()
Else : MessageBox.Show("Будь ласка, перевірте правильність введення фрази для навчання НМ",
"Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
End If
Catch ex As Exception
If ex.TargetSite.Name = "WinIOError" Then MsgBox("Файл зайнятий іншим додатком!",
MessageBoxIcon.Information)
End Try
End Sub

Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button6.Click
If Not System.IO.Directory.Exists(Direct & "\Навчання НМ") Then
System.IO.Directory.CreateDirectory(Direct & "\Навчання НМ")
Elseif (Not File.Exists(Direct & "\Навчання НМ\" & Label4.Text & "_train.txt")) Then
Dim g
g = File.Create(Direct & "\Навчання НМ\" & Label4.Text & "_train.txt")
g.Close()
End If
Try
Dim m4String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label4.Text & "_train.txt",
System.Text.Encoding.Default)
Dim b4 = m4String.Length
If TextBox4.Text = Label4.Text Then
tn1 = 0
tu1 = (ku1 - kd1).Milliseconds / 1000
tn2 = (kd2 - ku1).Milliseconds / 1000
tu2 = (ku2 - kd2).Milliseconds / 1000
tn3 = (kd3 - ku2).Milliseconds / 1000
tu3 = (ku3 - kd3).Milliseconds / 1000
tn4 = (kd4 - ku3).Milliseconds / 1000
tu4 = (ku4 - kd4).Milliseconds / 1000
tn5 = (kd5 - ku4).Milliseconds / 1000

```

```

tu5 = (ku5 - kd5).Milliseconds / 1000
tn6 = (kd6 - ku5).Milliseconds / 1000
tu6 = (ku6 - kd6).Milliseconds / 1000
tn7 = (kd7 - ku6).Milliseconds / 1000
tu7 = (ku7 - kd7).Milliseconds / 1000
tn8 = (kd8 - ku7).Milliseconds / 1000
tu8 = (ku8 - kd8).Milliseconds / 1000
tn9 = (kd9 - ku8).Milliseconds / 1000
tu9 = (ku9 - kd9).Milliseconds / 1000
tn10 = (kd10 - ku9).Milliseconds / 1000
tu10 = (ku10 - kd10).Milliseconds / 1000
Dim writer As New StreamWriter(Direct & "\Навчання НМ\" & Label4.Text & "_train.txt", True,
System.Text.Encoding.GetEncoding(1251))
writer.WriteLine(tn1 & vbTab & tu1 & vbTab & tn2 & vbTab & tu2 & vbTab & tn3 & vbTab & tu3 &
vbTab & tn4 & vbTab & tu4 & vbTab & tn5 & vbTab & tu5 & vbTab & tn6 & vbTab & tu6 & vbTab & tn7 & vbTab & tu7 &
vbTab & tn8 & vbTab & tu8 & vbTab & tn9 & vbTab & tu9 & vbTab & tn10 & vbTab & tu10) ' Записуємо в файл текст
writer.Close() ' Закриваємо файл
Dim Txt As String = IO.File.ReadAllText(Direct & "\Навчання НМ\" & Label4.Text & "_train.txt",
System.Text.Encoding.Default)
Txt = Txt.Replace(", ", ".")
IO.File.WriteAllText(Direct & "\Навчання НМ\" & Label4.Text & "_train.txt", Txt,
System.Text.Encoding.Default)

Label15.Text = "Кількість введень = " & b4 + 1
TextBox4.Text = ""
TextBox4.Select()
Else : MessageBox.Show("Будь ласка, перевірте правильність введення фрази для навчання НМ",
"Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
End If
Catch ex As Exception
If ex.TargetSite.Name = "WinIOError" Then MsgBox("Файл зайнятий іншим додатком!",
MessageBoxIcon.Information)
End Try
End Sub

Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button7.Click
If Not System.IO.Directory.Exists(Direct & "\Навчання НМ") Then
System.IO.Directory.CreateDirectory(Direct & "\Навчання НМ")
ElseIf (Not File.Exists(Direct & "\Навчання НМ\" & Label5.Text & "_train.txt")) Then
Dim g
g = File.Create(Direct & "\Навчання НМ\" & Label5.Text & "_train.txt")
g.Close()
End If
Try
Dim m5String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label5.Text & "_train.txt",
System.Text.Encoding.Default)
Dim b5 = m5String.Length
If TextBox5.Text = Label5.Text Then
tn1 = 0
tu1 = (ku1 - kd1).Milliseconds / 1000
tn2 = (kd2 - ku1).Milliseconds / 1000
tu2 = (ku2 - kd2).Milliseconds / 1000
tn3 = (kd3 - ku2).Milliseconds / 1000
tu3 = (ku3 - kd3).Milliseconds / 1000
tn4 = (kd4 - ku3).Milliseconds / 1000
tu4 = (ku4 - kd4).Milliseconds / 1000
tn5 = (kd5 - ku4).Milliseconds / 1000
tu5 = (ku5 - kd5).Milliseconds / 1000
tn6 = (kd6 - ku5).Milliseconds / 1000
tu6 = (ku6 - kd6).Milliseconds / 1000
tn7 = (kd7 - ku6).Milliseconds / 1000
tu7 = (ku7 - kd7).Milliseconds / 1000

```

```

tn8 = (kd8 - ku7).Milliseconds / 1000
tu8 = (ku8 - kd8).Milliseconds / 1000
tn9 = (kd9 - ku8).Milliseconds / 1000
tu9 = (ku9 - kd9).Milliseconds / 1000
tn10 = (kd10 - ku9).Milliseconds / 1000
tu10 = (ku10 - kd10).Milliseconds / 1000
Dim writer As New StreamWriter(Direct & "\Навчання НМ\" & Label5.Text & "_train.txt", True,
System.Text.Encoding.GetEncoding(1251))
writer.WriteLine(tn1 & vbTab & tu1 & vbTab & tn2 & vbTab & tu2 & vbTab & tn3 & vbTab & tu3 &
vbTab & tn4 & vbTab & tu4 & vbTab & tn5 & vbTab & tu5 & vbTab & tn6 & vbTab & tu6 & vbTab & tn7 & vbTab & tu7 &
vbTab & tn8 & vbTab & tu8 & vbTab & tn9 & vbTab & tu9 & vbTab & tn10 & vbTab & tu10) ' Записуємо в файл текст
writer.Close() ' Закриваємо файл
Dim Txt As String = IO.File.ReadAllText(Direct & "\Навчання НМ\" & Label5.Text & "_train.txt",
System.Text.Encoding.Default)
Txt = Txt.Replace(", ", ".")
IO.File.WriteAllText(Direct & "\Навчання НМ\" & Label5.Text & "_train.txt", Txt,
System.Text.Encoding.Default)

Label16.Text = "Кількість введень = " & b5 + 1
TextBox5.Text = ""
TextBox5.Select()
Else : MessageBox.Show("Будь ласка, перевірте правильність введення фрази для навчання НМ",
"Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
End If
Catch ex As Exception
If ex.TargetSite.Name = "WinIOError" Then MsgBox("Файл зайнятий іншим додатком!",
MessageBoxIcon.Information)
End Try
End Sub

Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button8.Click
If Not System.IO.Directory.Exists(Direct & "\Навчання НМ") Then
System.IO.Directory.CreateDirectory(Direct & "\Навчання НМ")
Elseif (Not File.Exists(Direct & "\Навчання НМ\" & Label6.Text & "_train.txt")) Then
Dim g
g = File.Create(Direct & "\Навчання НМ\" & Label6.Text & "_train.txt")
g.Close()
End If
Try
Dim m6String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label6.Text & "_train.txt",
System.Text.Encoding.Default)
Dim b6 = m6String.Length
If TextBox6.Text = Label6.Text Then
tn1 = 0
tu1 = (ku1 - kd1).Milliseconds / 1000
tn2 = (kd2 - ku1).Milliseconds / 1000
tu2 = (ku2 - kd2).Milliseconds / 1000
tn3 = (kd3 - ku2).Milliseconds / 1000
tu3 = (ku3 - kd3).Milliseconds / 1000
tn4 = (kd4 - ku3).Milliseconds / 1000
tu4 = (ku4 - kd4).Milliseconds / 1000
tn5 = (kd5 - ku4).Milliseconds / 1000
tu5 = (ku5 - kd5).Milliseconds / 1000
tn6 = (kd6 - ku5).Milliseconds / 1000
tu6 = (ku6 - kd6).Milliseconds / 1000
tn7 = (kd7 - ku6).Milliseconds / 1000
tu7 = (ku7 - kd7).Milliseconds / 1000
tn8 = (kd8 - ku7).Milliseconds / 1000
tu8 = (ku8 - kd8).Milliseconds / 1000
tn9 = (kd9 - ku8).Milliseconds / 1000
tu9 = (ku9 - kd9).Milliseconds / 1000
tn10 = (kd10 - ku9).Milliseconds / 1000

```

```

tu10 = (ku10 - kd10).Milliseconds / 1000
Dim writer As New StreamWriter(Direct & "\Навчання НМ\" & Label6.Text & "_train.txt", True,
System.Text.Encoding.GetEncoding(1251))
writer.WriteLine(tn1 & vbTab & tu1 & vbTab & tn2 & vbTab & tu2 & vbTab & tn3 & vbTab & tu3 &
vbTab & tn4 & vbTab & tu4 & vbTab & tn5 & vbTab & tu5 & vbTab & tn6 & vbTab & tu6 & vbTab & tn7 & vbTab & tu7 &
vbTab & tn8 & vbTab & tu8 & vbTab & tn9 & vbTab & tu9 & vbTab & tn10 & vbTab & tu10) ' Записуємо в файл текст
writer.Close() ' Закриваємо файл
Dim Txt As String = IO.File.ReadAllText(Direct & "\Навчання НМ\" & Label6.Text & "_train.txt",
System.Text.Encoding.Default)
Txt = Txt.Replace(", ", ".")
IO.File.WriteAllText(Direct & "\Навчання НМ\" & Label6.Text & "_train.txt", Txt,
System.Text.Encoding.Default)

Label17.Text = "Кількість введень = " & b6 + 1
TextBox6.Text = ""
TextBox6.Select()
Else : MessageBox.Show("Будь ласка, перевірте правильність введення фрази для навчання НМ",
"Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
End If
Catch ex As Exception
If ex.TargetSite.Name = "WinIOError" Then MsgBox("Файл зайнятий іншим додатком!",
MessageBoxIcon.Information)
End Try
End Sub

Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button9.Click
If Not System.IO.Directory.Exists(Direct & "\Навчання НМ") Then
System.IO.Directory.CreateDirectory(Direct & "\Навчання НМ")
ElseIf (Not File.Exists(Direct & "\Навчання НМ\" & Label7.Text & "_train.txt")) Then
Dim g
g = File.Create(Direct & "\Навчання НМ\" & Label7.Text & "_train.txt")
g.Close()
End If
Try
Dim m7String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label7.Text & "_train.txt",
System.Text.Encoding.Default)
Dim b7 = m7String.Length
If TextBox7.Text = Label7.Text Then
tn1 = 0
tu1 = (ku1 - kd1).Milliseconds / 1000
tn2 = (kd2 - ku1).Milliseconds / 1000
tu2 = (ku2 - kd2).Milliseconds / 1000
tn3 = (kd3 - ku2).Milliseconds / 1000
tu3 = (ku3 - kd3).Milliseconds / 1000
tn4 = (kd4 - ku3).Milliseconds / 1000
tu4 = (ku4 - kd4).Milliseconds / 1000
tn5 = (kd5 - ku4).Milliseconds / 1000
tu5 = (ku5 - kd5).Milliseconds / 1000
tn6 = (kd6 - ku5).Milliseconds / 1000
tu6 = (ku6 - kd6).Milliseconds / 1000
tn7 = (kd7 - ku6).Milliseconds / 1000
tu7 = (ku7 - kd7).Milliseconds / 1000
tn8 = (kd8 - ku7).Milliseconds / 1000
tu8 = (ku8 - kd8).Milliseconds / 1000
tn9 = (kd9 - ku8).Milliseconds / 1000
tu9 = (ku9 - kd9).Milliseconds / 1000
tn10 = (kd10 - ku9).Milliseconds / 1000
tu10 = (ku10 - kd10).Milliseconds / 1000
Dim writer As New StreamWriter(Direct & "\Навчання НМ\" & Label7.Text & "_train.txt", True,
System.Text.Encoding.GetEncoding(1251))

```

```

writer.WriteLine(tn1 & vbTab & tu1 & vbTab & tn2 & vbTab & tu2 & vbTab & tn3 & vbTab & tu3 &
vbTab & tn4 & vbTab & tu4 & vbTab & tn5 & vbTab & tu5 & vbTab & tn6 & vbTab & tu6 & vbTab & tn7 & vbTab & tu7 &
vbTab & tn8 & vbTab & tu8 & vbTab & tn9 & vbTab & tu9 & vbTab & tn10 & vbTab & tu10) ' Записуємо в файл текст
writer.Close() ' Закриваємо файл
Dim Txt As String = IO.File.ReadAllText(Direct & "\Навчання НМ\" & Label7.Text & "_train.txt",
System.Text.Encoding.Default)
Txt = Txt.Replace(", ", ".")
IO.File.WriteAllText(Direct & "\Навчання НМ\" & Label7.Text & "_train.txt", Txt,
System.Text.Encoding.Default)

Label18.Text = "Кількість введень = " & b7 + 1
TextBox7.Text = ""
TextBox7.Select()
Else : MessageBox.Show("Будь ласка, перевірте правильність введення фрази для навчання НМ",
"Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
End If
Catch ex As Exception
If ex.TargetSite.Name = "WinIOError" Then MsgBox("Файл зайнятий іншим додатком!",
MessageBoxIcon.Information)
End Try
End Sub

Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button10.Click
If Not System.IO.Directory.Exists(Direct & "\Навчання НМ") Then
System.IO.Directory.CreateDirectory(Direct & "\Навчання НМ")
ElseIf (Not File.Exists(Direct & "\Навчання НМ\" & Label8.Text & "_train.txt")) Then
Dim g
g = File.Create(Direct & "\Навчання НМ\" & Label8.Text & "_train.txt")
g.Close()
End If
Try
Dim m8String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label8.Text & "_train.txt",
System.Text.Encoding.Default)
Dim b8 = m8String.Length
If TextBox8.Text = Label8.Text Then
tn1 = 0
tu1 = (ku1 - kd1).Milliseconds / 1000
tn2 = (kd2 - ku1).Milliseconds / 1000
tu2 = (ku2 - kd2).Milliseconds / 1000
tn3 = (kd3 - ku2).Milliseconds / 1000
tu3 = (ku3 - kd3).Milliseconds / 1000
tn4 = (kd4 - ku3).Milliseconds / 1000
tu4 = (ku4 - kd4).Milliseconds / 1000
tn5 = (kd5 - ku4).Milliseconds / 1000
tu5 = (ku5 - kd5).Milliseconds / 1000
tn6 = (kd6 - ku5).Milliseconds / 1000
tu6 = (ku6 - kd6).Milliseconds / 1000
tn7 = (kd7 - ku6).Milliseconds / 1000
tu7 = (ku7 - kd7).Milliseconds / 1000
tn8 = (kd8 - ku7).Milliseconds / 1000
tu8 = (ku8 - kd8).Milliseconds / 1000
tn9 = (kd9 - ku8).Milliseconds / 1000
tu9 = (ku9 - kd9).Milliseconds / 1000
tn10 = (kd10 - ku9).Milliseconds / 1000
tu10 = (ku10 - kd10).Milliseconds / 1000
Dim writer As New StreamWriter(Direct & "\Навчання НМ\" & Label8.Text & "_train.txt", True,
System.Text.Encoding.GetEncoding(1251))
writer.WriteLine(tn1 & vbTab & tu1 & vbTab & tn2 & vbTab & tu2 & vbTab & tn3 & vbTab & tu3 &
vbTab & tn4 & vbTab & tu4 & vbTab & tn5 & vbTab & tu5 & vbTab & tn6 & vbTab & tu6 & vbTab & tn7 & vbTab & tu7 &
vbTab & tn8 & vbTab & tu8 & vbTab & tn9 & vbTab & tu9 & vbTab & tn10 & vbTab & tu10) ' Записуємо в файл текст
writer.Close() ' Закриваємо файл

```

```

        Dim Txt As String = IO.File.ReadAllText(Direct & "\Навчання НМ\" & Label8.Text & "_train.txt",
System.Text.Encoding.Default)
        Txt = Txt.Replace(",", ".")
        IO.File.WriteAllText(Direct & "\Навчання НМ\" & Label8.Text & "_train.txt", Txt,
System.Text.Encoding.Default)

        Label19.Text = "Кількість введень = " & b8 + 1
        TextBox8.Text = ""
        TextBox8.Select()
    Else : MessageBox.Show("Будь ласка, перевірте правильність введення фрази для навчання НМ",
"Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
    End If
    Catch ex As Exception
        If ex.TargetSite.Name = "WinIOError" Then MsgBox("Файл зайнятий іншим додатком!",
MessageBoxIcon.Information)
        End Try
    End Sub

    Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button11.Click
        If Not System.IO.Directory.Exists(Direct & "\Навчання НМ") Then
            System.IO.Directory.CreateDirectory(Direct & "\Навчання НМ")
        ElseIf (Not File.Exists(Direct & "\Навчання НМ\" & Label9.Text & "_train.txt")) Then
            Dim g
            g = File.Create(Direct & "\Навчання НМ\" & Label9.Text & "_train.txt")
            g.Close()
        End If
        Try
            Dim m9String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label9.Text & "_train.txt",
System.Text.Encoding.Default)
            Dim b9 = m9String.Length
            If TextBox9.Text = Label9.Text Then
                tn1 = 0
                tu1 = (ku1 - kd1).Milliseconds / 1000
                tn2 = (kd2 - ku1).Milliseconds / 1000
                tu2 = (ku2 - kd2).Milliseconds / 1000
                tn3 = (kd3 - ku2).Milliseconds / 1000
                tu3 = (ku3 - kd3).Milliseconds / 1000
                tn4 = (kd4 - ku3).Milliseconds / 1000
                tu4 = (ku4 - kd4).Milliseconds / 1000
                tn5 = (kd5 - ku4).Milliseconds / 1000
                tu5 = (ku5 - kd5).Milliseconds / 1000
                tn6 = (kd6 - ku5).Milliseconds / 1000
                tu6 = (ku6 - kd6).Milliseconds / 1000
                tn7 = (kd7 - ku6).Milliseconds / 1000
                tu7 = (ku7 - kd7).Milliseconds / 1000
                tn8 = (kd8 - ku7).Milliseconds / 1000
                tu8 = (ku8 - kd8).Milliseconds / 1000
                tn9 = (kd9 - ku8).Milliseconds / 1000
                tu9 = (ku9 - kd9).Milliseconds / 1000
                tn10 = (kd10 - ku9).Milliseconds / 1000
                tu10 = (ku10 - kd10).Milliseconds / 1000
                Dim writer As New StreamWriter(Direct & "\Навчання НМ\" & Label9.Text & "_train.txt", True,
System.Text.Encoding.GetEncoding(1251))
                writer.WriteLine(tn1 & vbTab & tu1 & vbTab & tn2 & vbTab & tu2 & vbTab & tn3 & vbTab & tu3 &
vbTab & tn4 & vbTab & tu4 & vbTab & tn5 & vbTab & tu5 & vbTab & tn6 & vbTab & tu6 & vbTab & tn7 & vbTab & tu7 &
vbTab & tn8 & vbTab & tu8 & vbTab & tn9 & vbTab & tu9 & vbTab & tn10 & vbTab & tu10) ' Записуємо в файл текст
                writer.Close() ' Закриваємо файл
                Dim Txt As String = IO.File.ReadAllText(Direct & "\Навчання НМ\" & Label9.Text & "_train.txt",
System.Text.Encoding.Default)
                Txt = Txt.Replace(",", ".")
                IO.File.WriteAllText(Direct & "\Навчання НМ\" & Label9.Text & "_train.txt", Txt,
System.Text.Encoding.Default)
            End Try
        End Sub
    End Sub

```

```

Label20.Text = "Кількість введень = " & b9 + 1
TextBox9.Text = ""
TextBox9.Select()
Else : MessageBox.Show("Будь ласка, перевірте правильність введення фрази для навчання НМ",
"Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
End If
Catch ex As Exception
If ex.TargetSite.Name = "WinIOError" Then MsgBox("Файл зайнятий іншим додатком!",
MessageBoxIcon.Information)
End Try
End Sub

Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button12.Click
If Not System.IO.Directory.Exists(Direct & "\Навчання НМ") Then
System.IO.Directory.CreateDirectory(Direct & "\Навчання НМ")
ElseIf (Not File.Exists(Direct & "\Навчання НМ\" & Label10.Text & "_train.txt")) Then
Dim g
g = File.Create(Direct & "\Навчання НМ\" & Label10.Text & "_train.txt")
g.Close()
End If
Try
Dim m10String() = IO.File.ReadAllLines(Direct & "\Навчання НМ\" & Label10.Text & "_train.txt",
System.Text.Encoding.Default)
Dim b10 = m10String.Length
If TextBox10.Text = Label10.Text Then
tn1 = 0
tu1 = (ku1 - kd1).Milliseconds / 1000
tn2 = (kd2 - ku1).Milliseconds / 1000
tu2 = (ku2 - kd2).Milliseconds / 1000
tn3 = (kd3 - ku2).Milliseconds / 1000
tu3 = (ku3 - kd3).Milliseconds / 1000
tn4 = (kd4 - ku3).Milliseconds / 1000
tu4 = (ku4 - kd4).Milliseconds / 1000
tn5 = (kd5 - ku4).Milliseconds / 1000
tu5 = (ku5 - kd5).Milliseconds / 1000
tn6 = (kd6 - ku5).Milliseconds / 1000
tu6 = (ku6 - kd6).Milliseconds / 1000
tn7 = (kd7 - ku6).Milliseconds / 1000
tu7 = (ku7 - kd7).Milliseconds / 1000
tn8 = (kd8 - ku7).Milliseconds / 1000
tu8 = (ku8 - kd8).Milliseconds / 1000
tn9 = (kd9 - ku8).Milliseconds / 1000
tu9 = (ku9 - kd9).Milliseconds / 1000
tn10 = (kd10 - ku9).Milliseconds / 1000
tu10 = (ku10 - kd10).Milliseconds / 1000
Dim writer As New StreamWriter(Direct & "\Навчання НМ\" & Label10.Text & "_train.txt", True,
System.Text.Encoding.GetEncoding(1251))
writer.WriteLine(tn1 & vbTab & tu1 & vbTab & tn2 & vbTab & tu2 & vbTab & tn3 & vbTab & tu3 &
vbTab & tn4 & vbTab & tu4 & vbTab & tn5 & vbTab & tu5 & vbTab & tn6 & vbTab & tu6 & vbTab & tn7 & vbTab & tu7 &
vbTab & tn8 & vbTab & tu8 & vbTab & tn9 & vbTab & tu9 & vbTab & tn10 & vbTab & tu10) ' Записуємо в файл текст
writer.Close() ' Закриваємо файл
Dim Txt As String = IO.File.ReadAllText(Direct & "\Навчання НМ\" & Label10.Text & "_train.txt",
System.Text.Encoding.Default)
Txt = Txt.Replace(", ", ".")
IO.File.WriteAllText(Direct & "\Навчання НМ\" & Label10.Text & "_train.txt", Txt,
System.Text.Encoding.Default)

Label21.Text = "Кількість введень = " & b10 + 1
TextBox10.Text = ""
TextBox10.Select()

```



```

        Else : MessageBox.Show("Будь ласка, перевірте правильність введення фрази для навчання НМ",
"Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
    End If
    Catch ex As Exception
        If ex.TargetSite.Name = "WinIOError" Then MsgBox("Файл зайнятий іншим додатком!",
MessageBoxIcon.Information)
    End Try
End Sub

Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button13.Click
    Form5.Visible = True
    Me.Visible = False
End Sub
End Class

```

Код для «Перевірка поточного ФС» (Form2)

```

Imports System.IO
Imports System.Runtime.InteropServices
Imports Microsoft.Office.Interop.Excel

Public Class Form2
    Dim kd1, kd2, kd3, kd4, kd5, kd6, kd7, kd8, kd9, kd10 As Date 'событие нажатия на кнопку
    Dim ku1, ku2, ku3, ku4, ku5, ku6, ku7, ku8, ku9, ku10 As Date 'событие отпуска кнопки
    Dim tn1, tn2, tn3, tn4, tn5, tn6, tn7, tn8, tn9, tn10 As Single 'время между нажатиями кнопок
    Dim tu1, tu2, tu3, tu4, tu5, tu6, tu7, tu8, tu9, tu10 As Single 'время удержания кнопок
    Dim reader As StreamReader
    Dim Words() As String = {"адекватний", "бібліотека", "відеопотік", "гермомішок", "дендропарк",
"зубочистка", "катастрофа", "латифундія", "менеджмент", "наповнення"}
    Dim rnd As Random = New Random

    Private Function GetNext(ByVal S() As String) As String
        Return S(rnd.Next(S.Count))
    End Function

    Private Sub TextBox1_KeyDown(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyDown
        If Label1.Text = "адекватний" Then
            If e.KeyCode = 70 Then
                kd1 = Now()
            End If
            If e.KeyCode = 76 Then
                kd2 = Now()
            End If
            If e.KeyCode = 84 Then
                kd3 = Now()
            End If
            If e.KeyCode = 82 Then
                kd4 = Now()
            End If
            If e.KeyCode = 68 Then
                kd5 = Now()
            End If
            If e.KeyCode = 70 Then
                kd6 = Now()
            End If
            If e.KeyCode = 78 Then
                kd7 = Now()
            End If

```

```

    If e.KeyCode = 89 Then
        kd8 = Now()
    End If
    If e.KeyCode = 66 Then
        kd9 = Now()
    End If
    If e.KeyCode = 81 Then
        kd10 = Now()
    End If
End If

```

```

.....
End Sub

```

```

Private Sub TextBox1_KeyUp(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox1.KeyUp

```

```

    If Label1.Text = "адекватный" Then

```

```

        If e.KeyCode = 70 Then
            ku1 = Now()

```

```

        End If

```

```

        If e.KeyCode = 76 Then
            ku2 = Now()

```

```

        End If

```

```

        If e.KeyCode = 84 Then
            ku3 = Now()

```

```

        End If

```

```

        If e.KeyCode = 82 Then
            ku4 = Now()

```

```

        End If

```

```

        If e.KeyCode = 68 Then
            ku5 = Now()

```

```

        End If

```

```

        If e.KeyCode = 70 Then
            ku6 = Now()

```

```

        End If

```

```

        If e.KeyCode = 78 Then
            ku7 = Now()

```

```

        End If

```

```

        If e.KeyCode = 89 Then
            ku8 = Now()

```

```

        End If

```

```

        If e.KeyCode = 66 Then
            ku9 = Now()

```

```

        End If

```

```

        If e.KeyCode = 81 Then
            ku10 = Now()

```

```

        End If

```

```

    End If

```

```

.....
End Sub

```

```

Private Sub someButton_MouseHover(ByVal sender As Object, ByVal e As System.EventArgs) _

```

```

    Handles Button4.MouseHover

```

```

    Dim toolTip As New ToolTip()

```

```

    toolTip.SetToolTip(Button3, "На голову")

```

```

End Sub

```

```

Private Sub Form2_FormClosed(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.FormClosedEventArgs) Handles MyBase.FormClosed

```

```

    Me.Visible = False

```

```

    Form4.Visible = True

```

```

End Sub

```

```

Private Sub Form1_Shown(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Shown
    Label1.Text = GetNext(Words)
End Sub

Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As KeyPressEventArgs) Handles
TextBox1.KeyPress
    If Not Char.IsLetter(e.KeyChar) And Not Char.IsControl(e.KeyChar) Then
        e.KeyChar = ""
    End If
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
    Dim xlsApp As Object
    Dim xlsWb As Object
    Dim xlsSh As Object
    If TextBox1.Text = Label1.Text Then
        tn1 = 0
        tu1 = (ku1 - kd1).Milliseconds / 1000
        tn2 = (kd2 - ku1).Milliseconds / 1000
        tu2 = (ku2 - kd2).Milliseconds / 1000
        tn3 = (kd3 - ku2).Milliseconds / 1000
        tu3 = (ku3 - kd3).Milliseconds / 1000
        tn4 = (kd4 - ku3).Milliseconds / 1000
        tu4 = (ku4 - kd4).Milliseconds / 1000
        tn5 = (kd5 - ku4).Milliseconds / 1000
        tu5 = (ku5 - kd5).Milliseconds / 1000
        tn6 = (kd6 - ku5).Milliseconds / 1000
        tu6 = (ku6 - kd6).Milliseconds / 1000
        tn7 = (kd7 - ku6).Milliseconds / 1000
        tu7 = (ku7 - kd7).Milliseconds / 1000
        tn8 = (kd8 - ku7).Milliseconds / 1000
        tu8 = (ku8 - kd8).Milliseconds / 1000
        tn9 = (kd9 - ku8).Milliseconds / 1000
        tu9 = (ku9 - kd9).Milliseconds / 1000
        tn10 = (kd10 - ku9).Milliseconds / 1000
        tu10 = (ku10 - kd10).Milliseconds / 1000

        If Not System.IO.Directory.Exists(Direct & "\Поточный ФС") Then
            System.IO.Directory.CreateDirectory(Direct & "\Поточный ФС")
        ElseIf (Not File.Exists(Direct & "\Поточный ФС\ФС_now.txt")) Then
            Dim g
            g = File.Create(Direct & "\Поточный ФС\ФС_now.txt")
            g.Close()
        End If
        If Not System.IO.Directory.Exists(Direct & "\Результат") Then
            System.IO.Directory.CreateDirectory(Direct & "\Результат")
        ElseIf (Not File.Exists(Direct & "\Результат\Result.txt")) Then
            Dim g
            g = File.Create(Direct & "\Результат\Result.txt")
            g.Close()
        End If

        Dim writer As New StreamWriter(Direct & "\Поточный ФС\ФС_now.txt", False,
System.Text.Encoding.GetEncoding(1251))
        writer.WriteLine(tn1)
        writer.WriteLine(tu1)
        writer.WriteLine(tn2)
        writer.WriteLine(tu2)
        writer.WriteLine(tn3)
        writer.WriteLine(tu3)
        writer.WriteLine(tn4)

```

```

writer.WriteLine(tu4)
writer.WriteLine(tn5)
writer.WriteLine(tu5)
writer.WriteLine(tn6)
writer.WriteLine(tu6)
writer.WriteLine(tn7)
writer.WriteLine(tu7)
writer.WriteLine(tn8)
writer.WriteLine(tu8)
writer.WriteLine(tn9)
writer.WriteLine(tu9)
writer.WriteLine(tn10)
writer.WriteLine(tu10)
writer.Close()
Dim Txt As String = IO.File.ReadAllText(Direct & "\\Поточний ФС\ФС_now.txt",
System.Text.Encoding.Default)
Txt = Txt.Replace(", ", ".")
IO.File.WriteAllText(Direct & "\\Поточний ФС\ФС_now.txt", Txt, System.Text.Encoding.Default)

Dim ТипМатЛаб = Type.GetTypeFromProgID("Matlab.Application")
Dim МатЛаб = Activator.CreateInstance(ТипМатЛаб)
Dim now As String = "data = load (" & Direct & "\\Поточний ФС\ФС_now.txt');"
Dim result As String = "dlmwrite(" & Direct & "\\Результат\Result.txt',a(1,1))"
МатЛаб.visible = False
If Label1.Text = "адекватний" Then
    Dim train As String = "D = load (" & Direct & "\\Навчання НМ\адекватний_train.txt');"
    Dim Команди = {train + "net=newcD;" + "net = newsom(P,[1 2]);" + "net = train(net, P);" + now + "a =
sim(net, data);" + result}
    Dim Результат As String = ТипМатЛаб.InvokeMember("Execute",
Reflection.BindingFlags.InvokeMethod, Nothing, МатЛаб, Команди)
End If
If Label1.Text = "бібліотека" Then
    Dim train As String = "D = load (" & Direct & "\\Навчання НМ\бібліотека_train.txt');"
    Dim Команди = {train + "P = D;" + "net = newsom(P,[1 2]);" + "net = train(net, P);" + now + "a = sim(net,
data);" + result}
    Dim Результат As String = ТипМатЛаб.InvokeMember("Execute",
Reflection.BindingFlags.InvokeMethod, Nothing, МатЛаб, Команди)
End If
If Label1.Text = "відеопотік" Then
    Dim train As String = "D = load (" & Direct & "\\Навчання НМ\відеопотік_train.txt');"
    Dim Команди = {train + "P = D;" + "net = newsom(P,[1 2]);" + "net = train(net, P);" + now + "a = sim(net,
data);" + result}
    Dim Результат As String = ТипМатЛаб.InvokeMember("Execute",
Reflection.BindingFlags.InvokeMethod, Nothing, МатЛаб, Команди)
End If
If Label1.Text = "гермомішок" Then
    Dim train As String = "D = load (" & Direct & "\\Навчання НМ\гермомішок_train.txt');"
    Dim Команди = {train + "P = D;" + "net = newsom(P,[1 2]);" + "net = train(net, P);" + now + "a = sim(net,
data);" + result}
    Dim Результат As String = ТипМатЛаб.InvokeMember("Execute",
Reflection.BindingFlags.InvokeMethod, Nothing, МатЛаб, Команди)
End If
If Label1.Text = "дендропарк" Then
    Dim train As String = "D = load (" & Direct & "\\Навчання НМ\дендропарк_train.txt');"
    Dim Команди = {train + "P = D;" + "net = newsom(P,[1 2]);" + "net = train(net, P);" + now + "a = sim(net,
data);" + result}
    Dim Результат As String = ТипМатЛаб.InvokeMember("Execute",
Reflection.BindingFlags.InvokeMethod, Nothing, МатЛаб, Команди)
End If
If Label1.Text = "зубочистка" Then
    Dim train As String = "D = load (" & Direct & "\\Навчання НМ\зубочистка_train.txt');"
    Dim Команди = {train + "P = D;" + "net = newsom(P,[1 2]);" + "net = train(net, P);" + now + "a = sim(net,
data);" + result}

```

```

        Dim Результат As String = ТипМатЛаб.InvokeMember("Execute",
Reflection.BindingFlags.InvokeMethod, Nothing, МатЛаб, Команди)
    End If
    If Label1.Text = "катастрофа" Then
        Dim train As String = "D = load (" & Direct & "\Навчання НМ\катастрофа_train.txt');"
        Dim Команди = {train + "P = D';" + "net = newsom(P,[1 2]);" + "net = train(net, P);" + now + "a = sim(net,
data);" + result}
        Dim Результат As String = ТипМатЛаб.InvokeMember("Execute",
Reflection.BindingFlags.InvokeMethod, Nothing, МатЛаб, Команди)
    End If
    If Label1.Text = "латифундія" Then
        Dim train As String = "D = load (" & Direct & "\Навчання НМ\латифундія_train.txt');"
        Dim Команди = {train + "P = D';" + "net = newsom(P,[1 2]);" + "net = train(net, P);" + now + "a = sim(net,
data);" + result}
        Dim Результат As String = ТипМатЛаб.InvokeMember("Execute",
Reflection.BindingFlags.InvokeMethod, Nothing, МатЛаб, Команди)
    End If
    If Label1.Text = "менеджмент" Then
        Dim train As String = "D = load (" & Direct & "\Навчання НМ\менеджмент_train.txt');"
        Dim Команди = {train + "P = D';" + "net = newsom(P,[1 2]);" + "net = train(net, P);" + now + "a = sim(net,
data);" + result}
        Dim Результат As String = ТипМатЛаб.InvokeMember("Execute",
Reflection.BindingFlags.InvokeMethod, Nothing, МатЛаб, Команди)
    End If
    If Label1.Text = "наповнення" Then
        Dim train As String = "D = load (" & Direct & "\Навчання НМ\наповнення_train.txt');"
        Dim Команди = {train + "P = D';" + "net = newsom(P,[1 2]);" + "net = train(net, P);" + now + "a = sim(net,
data);" + result}
        Dim Результат As String = ТипМатЛаб.InvokeMember("Execute",
Reflection.BindingFlags.InvokeMethod, Nothing, МатЛаб, Команди)
    End If

    Dim Txt1 As String = IO.File.ReadAllText(Direct & "\Результат\Result.txt", System.Text.Encoding.Default)
    If Txt1 = 0 Then
        MessageBox.Show("ЗАДОВІЛЬНИЙ функціональний стан! (*Бажаємо успіху в подальшій роботі*)",
"Повідомлення!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
    ElseIf Txt1 = 1 Then
        MessageBox.Show("НЕЗАДОВІЛЬНИЙ функціональний стан! (*Відпочиньте та повертайтеся до
роботи*)", "Повідомлення!", MessageBoxButtons.OK, MessageBoxIcon.Stop)
    End If

    xlsApp = CreateObject("Excel.Application")
    If (Not File.Exists(Direct & "\Результат\Статистика.xls")) Then
        Dim g
        g = File.Create(Direct & "\Результат\Статистика.xls")
        g.Close()
    End If
    xlsWb = xlsApp.WorkBooks.Open(Direct & "\Результат\Статистика.xls")
    xlsSh = xlsWb.Sheets(1)
    Dim i = 2
    Do
        If xlsSh.Cells(i, 1).Value = "" Then
            xlsSh.Cells(i, 1).Value = CStr(DateTime.Now)
            Exit Do
        End If
        i = i + 1
    Loop
    i = 2
    Do
        If xlsSh.Cells(i, 2).Value = "" Then
            If Txt1 = 0 Then
                xlsSh.Cells(i, 2).Value = "Задовільний функціональний стан"
            ElseIf Txt1 = 1 Then

```

```

        xlsSh.cells(i, 2).Value = "Незадовільний функціональний стан"
    End If
    Exit Do
End If
i = i + 1
Loop
i = 2
Do
    If xlsSh.cells(i, 3).Value = "" Then
        xlsSh.cells(i, 3).Value = Label1.Text
    Exit Do
    End If
    i = i + 1
Loop
xlsWb.Close(True)
xlsApp.Quit()
Else : MessageBox.Show("Будь ласка, перевірте правильність введення даного слова!", "Помилка!",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
End If
Label1.Text = GetNext(Words)
TextBox1.Text = ""
TextBox1.Select()
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
    If Dir(Direct & "\Результат\Статистика.xls") = "" Then
        MsgBox("Файлу не існує!", MessageBoxIcon.Warning)
    Else
        Process.Start(Direct & "\Результат\Статистика.xls")
    End If
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
    Try
        If Dir(Direct & "\Результат\Статистика.xls") = "" Then
            MsgBox("Файлу не існує!", MessageBoxIcon.Warning)
        Else
            File.Delete(Direct & "\Результат\Статистика.xls")
            MsgBox("Файл видалено!", MessageBoxIcon.Information)
        End If
    Catch ex As Exception
        If ex.TargetSite.Name = "WinIOError" Then MsgBox("Файл зайнятий іншим додатком!",
MessageBoxIcon.Information)
    End Try
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
    Me.Visible = False
    Form3.Visible = True
End Sub
End Class

```

Код для «Видалення файлів із вхідними даними» (Form5)

```
Imports System.IO
```

```
Public Class Form5
```

```
    Private Sub Form5_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```

For Each foundFile As String In My.Computer.FileSystem.GetFiles(Direct & "\Навчання НМ\")
    ComboBox1.Items.Add(foundFile)
Next
End Sub

```

```

Private Sub someButton_MouseHover(ByVal sender As Object, ByVal e As System.EventArgs) _
    Handles Button1.MouseHover
    Dim toolTip As New ToolTip()
    toolTip.SetToolTip(Button1, "Видалити вибраний файл")
    toolTip.SetToolTip(Button2, "Видалити всі файли")
End Sub

```

```

Private Sub Form5_Closing(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.FormClosing
    Form1.Visible = True
End Sub

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click 'one
    If ComboBox1.SelectedIndex = -1 Then
        MsgBox("Оберіть файл з випадного списку!", MessageBoxIcon.Warning)
    Else : File.Delete(ComboBox1.Text.ToString())
        ComboBox1.Items.Clear()
        For Each foundFile As String In My.Computer.FileSystem.GetFiles(Direct & "\Навчання НМ\")
            ComboBox1.Items.Add(foundFile)
        Next
        ComboBox1.Text = ""
        MsgBox("Файл " & ComboBox1.Text.ToString() & " видалено!", MessageBoxIcon.Information)
    End If
End Sub

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click 'all
    If CreateObject("Scripting.FileSystemObject").FolderExists(Direct & "\Навчання НМ") Then
        If CreateObject("Scripting.FileSystemObject").GetFolder(Direct & "\Навчання НМ").Files.Count > 0 Then
            For Each txt$ In IO.Directory.GetFiles(Direct & "\Навчання НМ", "*.txt")
                IO.File.Delete(txt)
            Next
            ComboBox1.Items.Clear()
            For Each foundFile As String In My.Computer.FileSystem.GetFiles(Direct & "\Навчання НМ\")
                ComboBox1.Items.Add(foundFile)
            Next
            MsgBox("Файли видалено!", MessageBoxIcon.Information)
        Else : MsgBox("Папка " & Direct & "\Навчання НМ" & " пуста!", MessageBoxIcon.Warning)
        End If
    Else : MsgBox("Папки " & Direct & "\Навчання НМ" & " не існує!", MessageBoxIcon.Warning)
    End If
End Sub
End Class

```

Код для «Автор» (Form6)

```

Public Class Form6
    Private Sub Form6_Closed(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.FormClosed
        Form3.Visible = True
    End Sub
End Class

```

ДОДАТОК Г. Інструкція з використання інформаційної системи

Для коректної роботи програми «Визначення ФС» необхідно виконати дві дії:

1. Встановити Matlab R2011b або вище.
2. Встановити програму «Визначення ФС» за допомогою файлу «Setup_FS.msi».

Установка інформаційної системи на комп'ютер

Якщо на комп'ютері оператора не встановлений Matlab R2011b або вище, тоді слід його встановити [36].

Далі необхідно зберегти пакет установника Windows (.msi) у зручному для користувача місці. Наприклад, як показано на рис. Г.1.

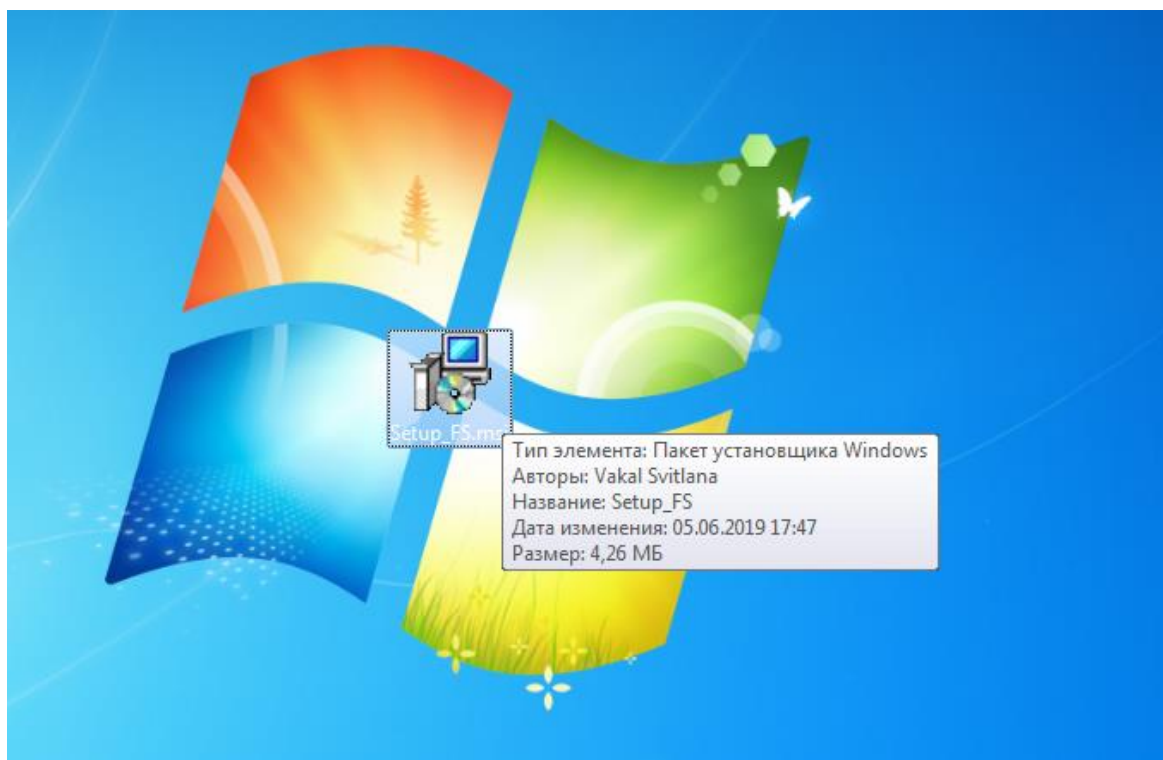


Рисунок Г.1 – Пакет установника знаходиться на Робочому столі

Натиснувши на нього двічі лівою клавiшею мишки або вибравши команду у випадному вікні «Установити» встановити систему на комп'ютер. Після виконання даної дії відкривається стартове вікно встановлення програми для оцінки функціонального стану операторів (рис. Г.2).

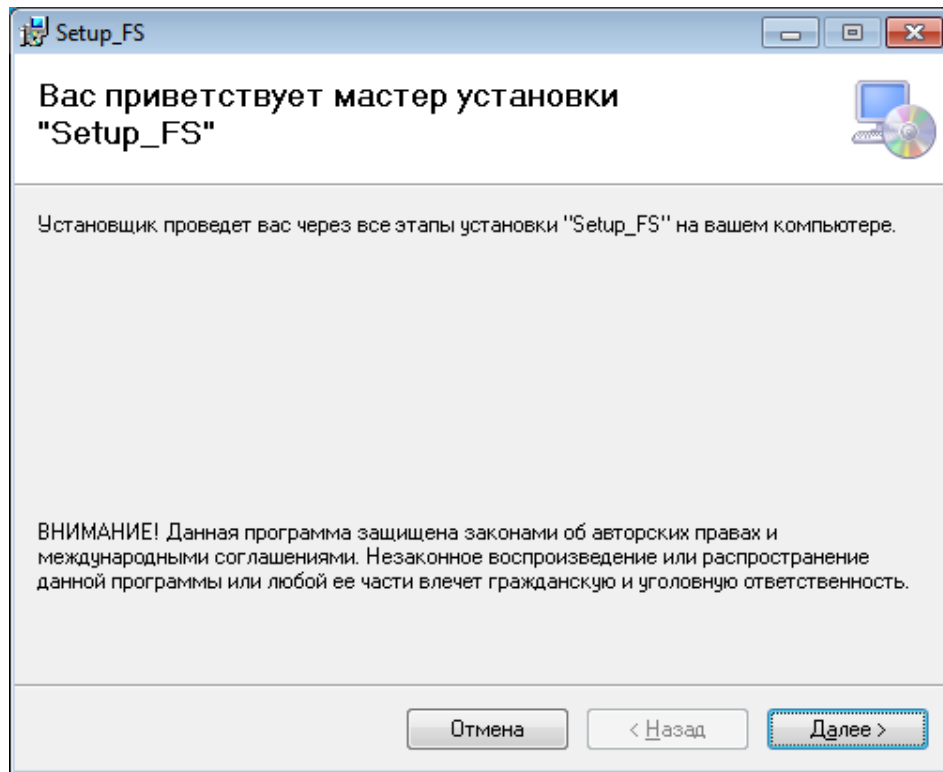


Рисунок Г.2 – Стартове вікно встановлення програми «Визначення ФС»

Натиснувши кнопку «Далее» користувач має змогу обрати до якої з директорій встановити інформаційну систему (рис. Г.3).

Обравши папку натиснути «Далее» та підтвердити установку програми на комп'ютер оператора (рис. Г.4).

Далі відбувається установка програми «Визначення ФС» (рис. Г.5) і по завершенню відкривається вікно з інформацією про успішне або неуспішне завершення роботи майстра установки (рис. Г.6).

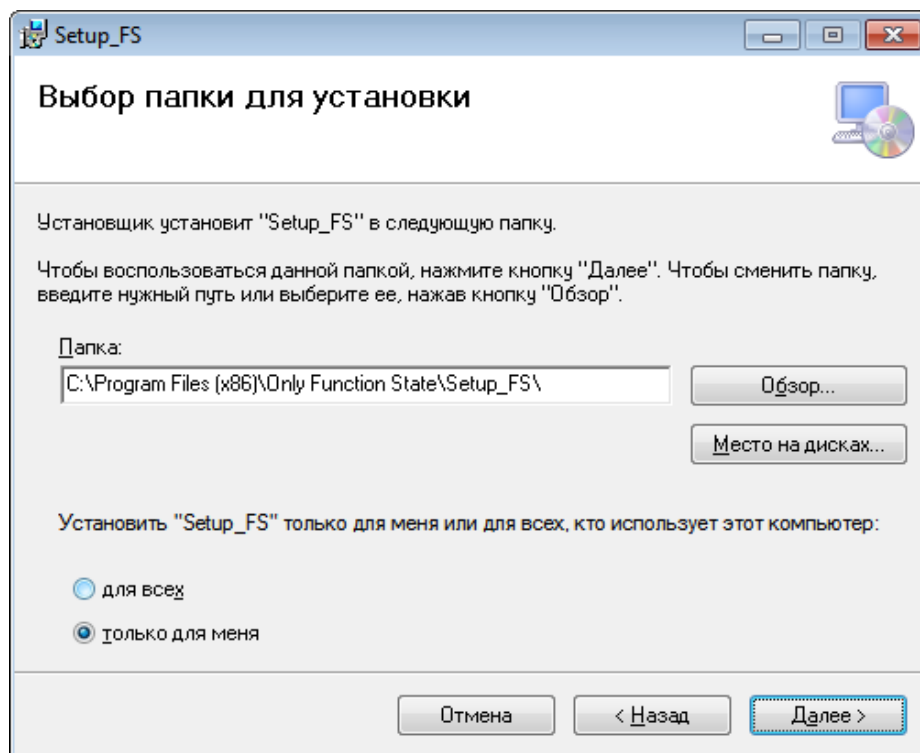


Рисунок Г.3 – Вибір директорії для встановлення програми «Визначення ФС»

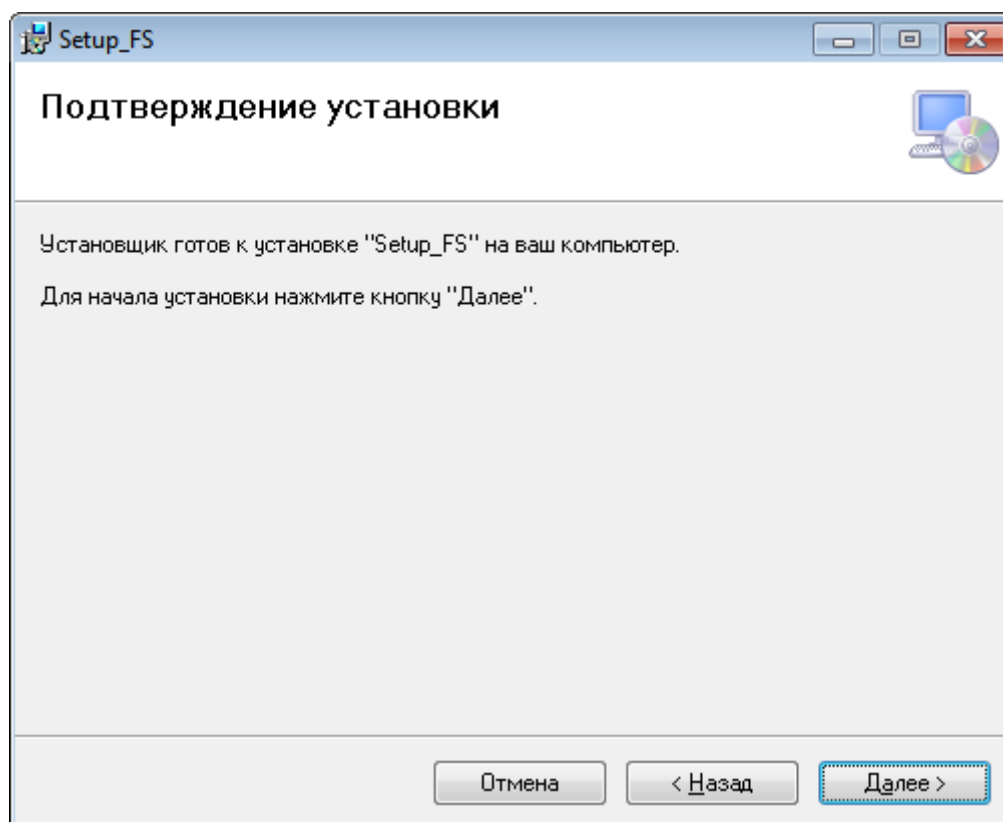


Рисунок Г.4 – Підтвердження встановлення програми «Визначення ФС»

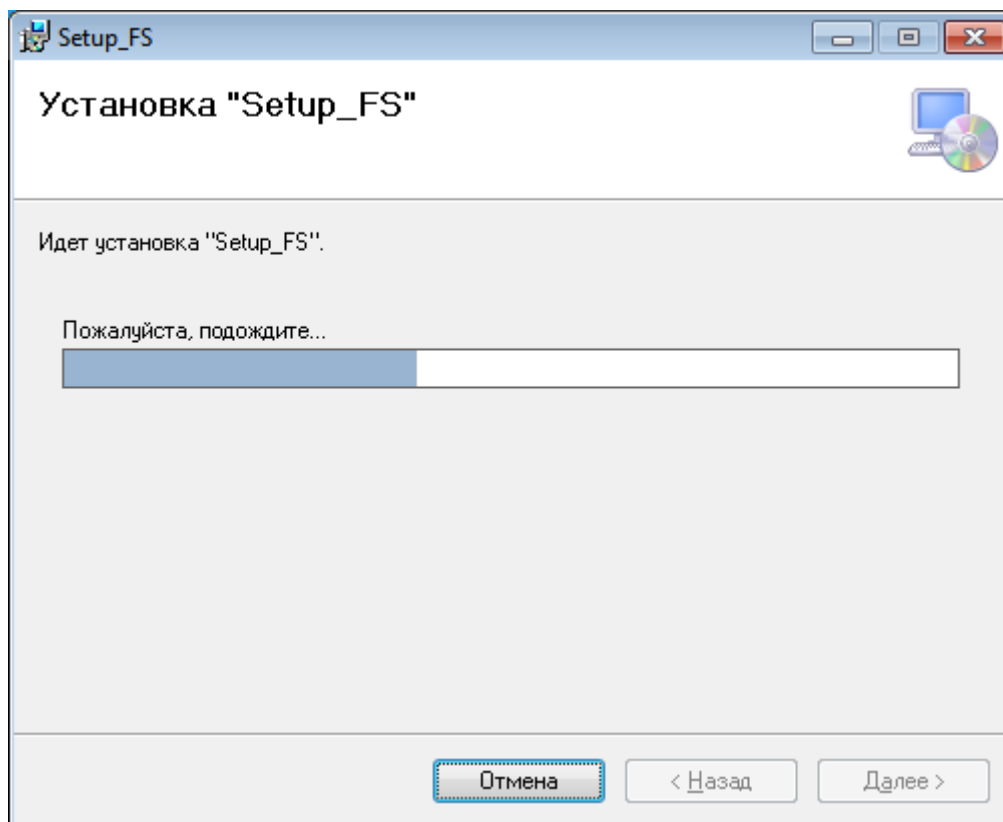


Рисунок Г.5 – Установка программы «Визначення ФС»

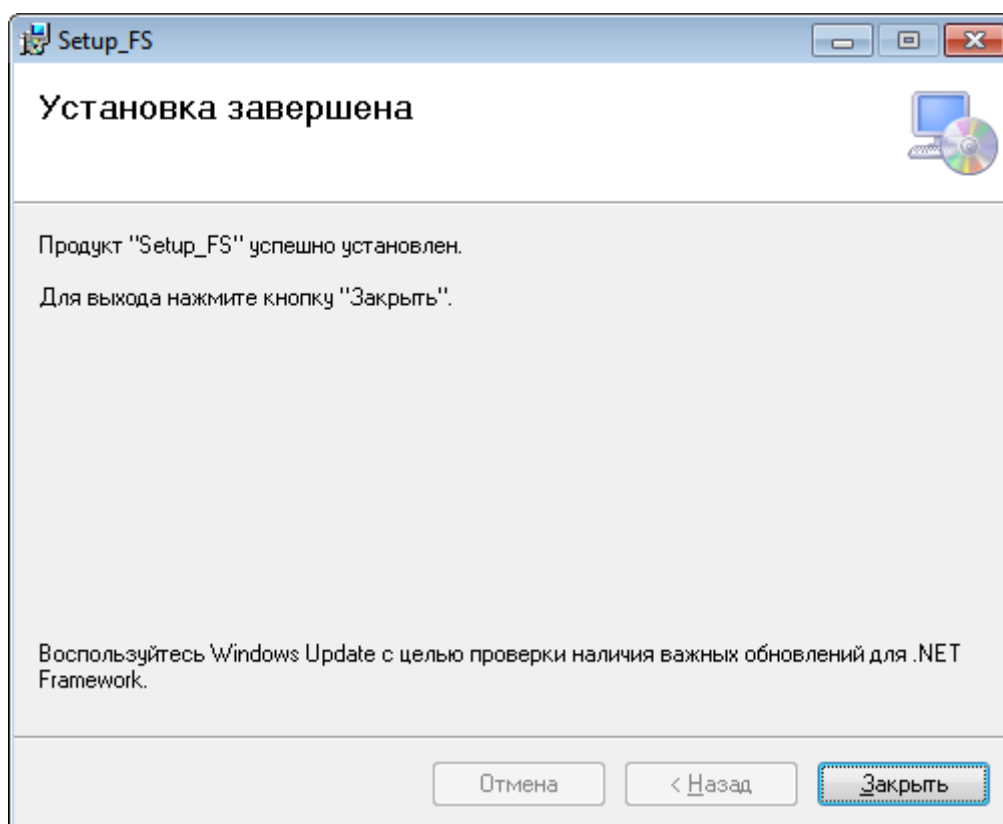


Рисунок Г.6 – Успішне завершення установки

У результаті установки на робочому столі з'явиться ярлик програми «Визначення ФС» (рис. Г.7).



Рисунок Г.7 – Ярлик програми на робочому столі

Робота з програмою «Визначення ФС»

Запустити програму можна натиснувши двічі на ярлик програми «Визначення ФС». У результаті програму запущено, користувач бачить головне вікно програми (рис. Г.8).

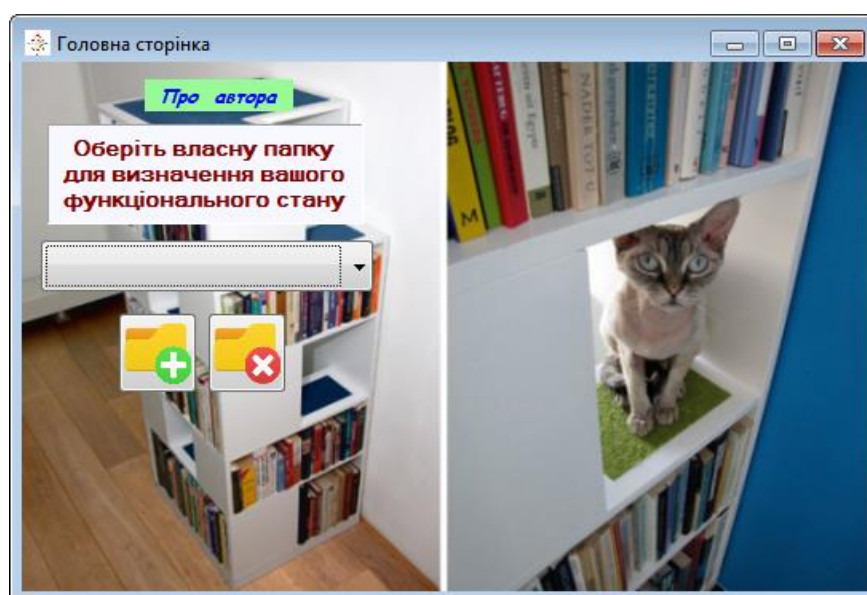


Рисунок Г.8 – Головне вікно програми

На вікні активні три кнопки: «Про автора», додавання та видалення папки. При натисканні відбуваються наступні дії, які показані на рис. Г.9 – Г.11.

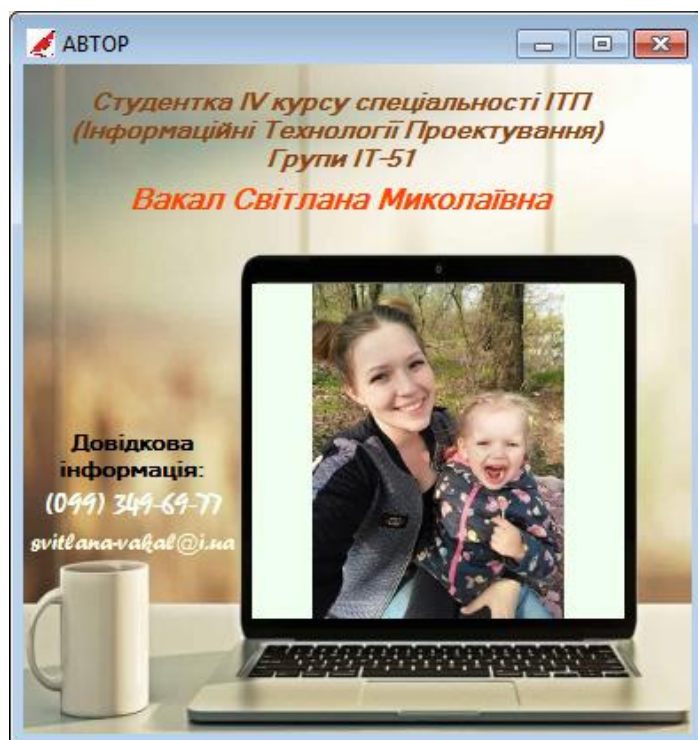


Рисунок Г.9 – Вікно «Автор», після натискання «Про автора» на головному вікні програми

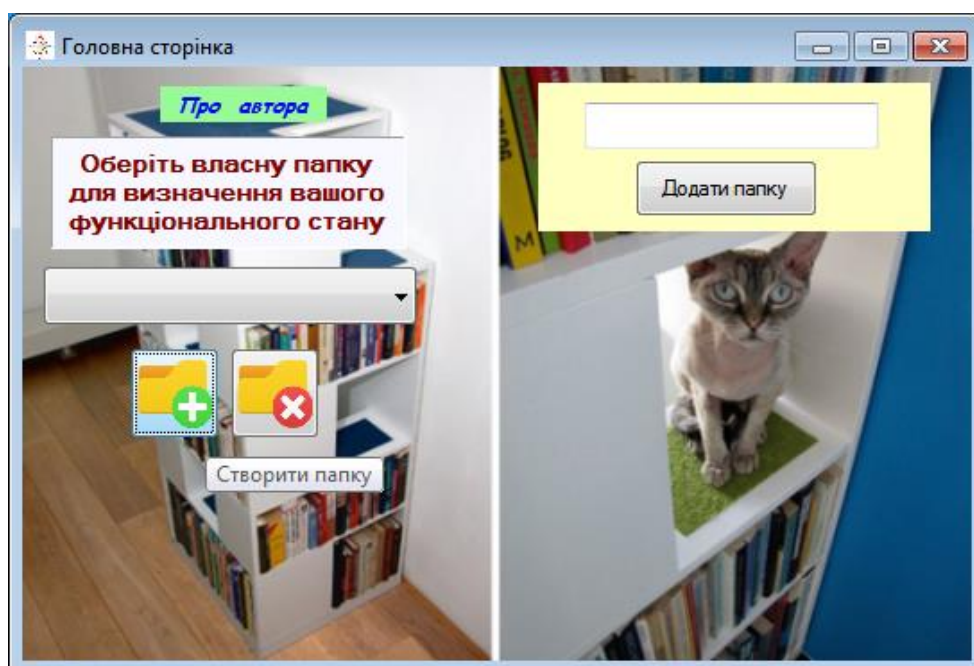


Рисунок Г.10 – Активна можливість створення папки

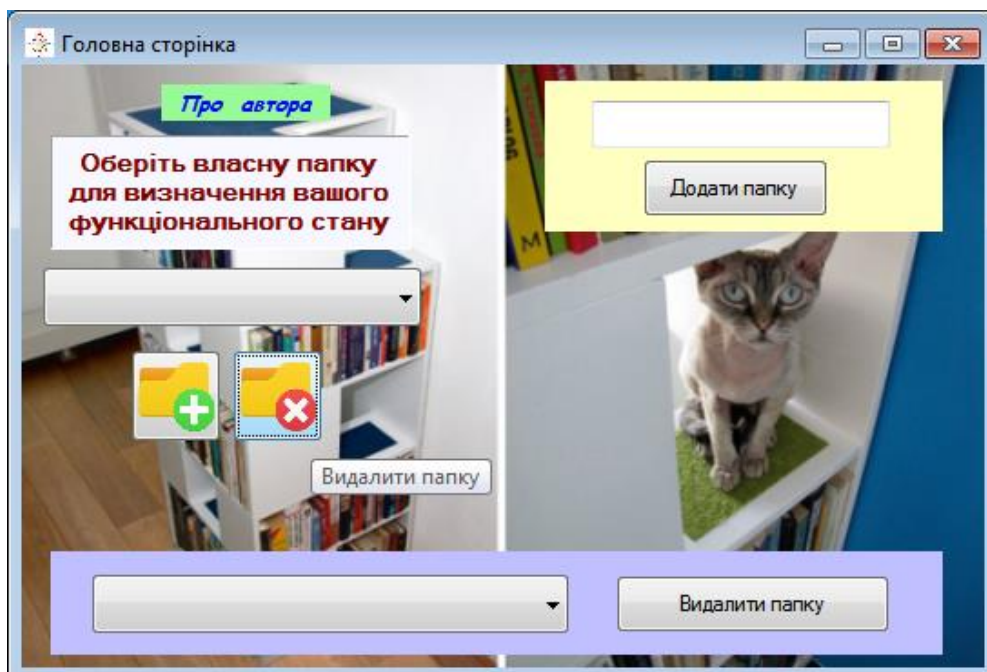


Рисунок Г.11 – Активна можливість видалення папки

Обираючи папку (рис. Г.12) користувач входить до директорії власної папки та має змогу виконати 2 дії (рис. Г.13), в залежності від того чи створив він навчальну вибірку чи ні.

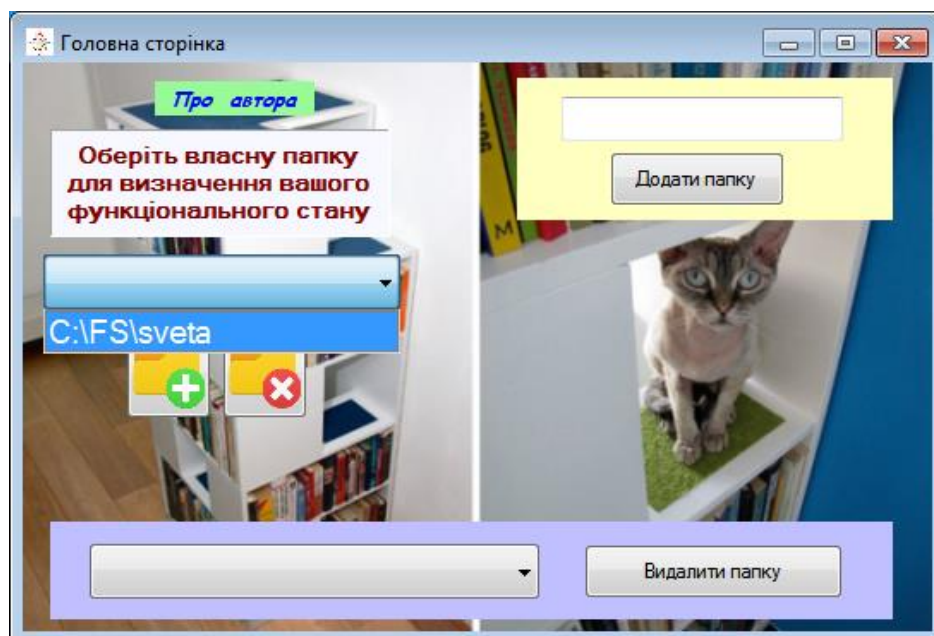


Рисунок Г.12 – Вибір папки

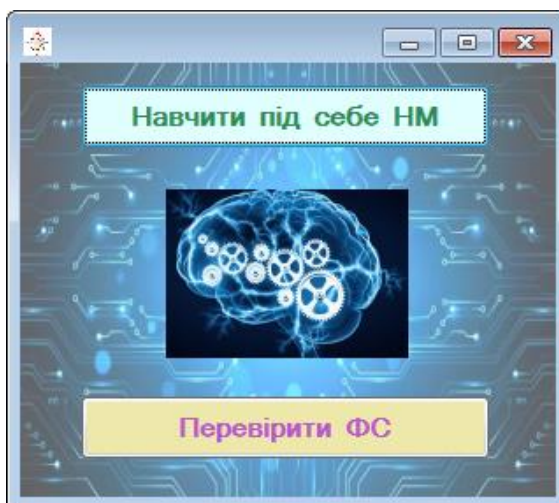


Рисунок Г.13 – Проміжне вікно вибору дії

Натиснувши на «Навчити під себе НМ» користувач опиняється на вікні «Персональні налаштування системи» (рис. Г.14). У цьому вікні відбувається створення навчальної вибірки.

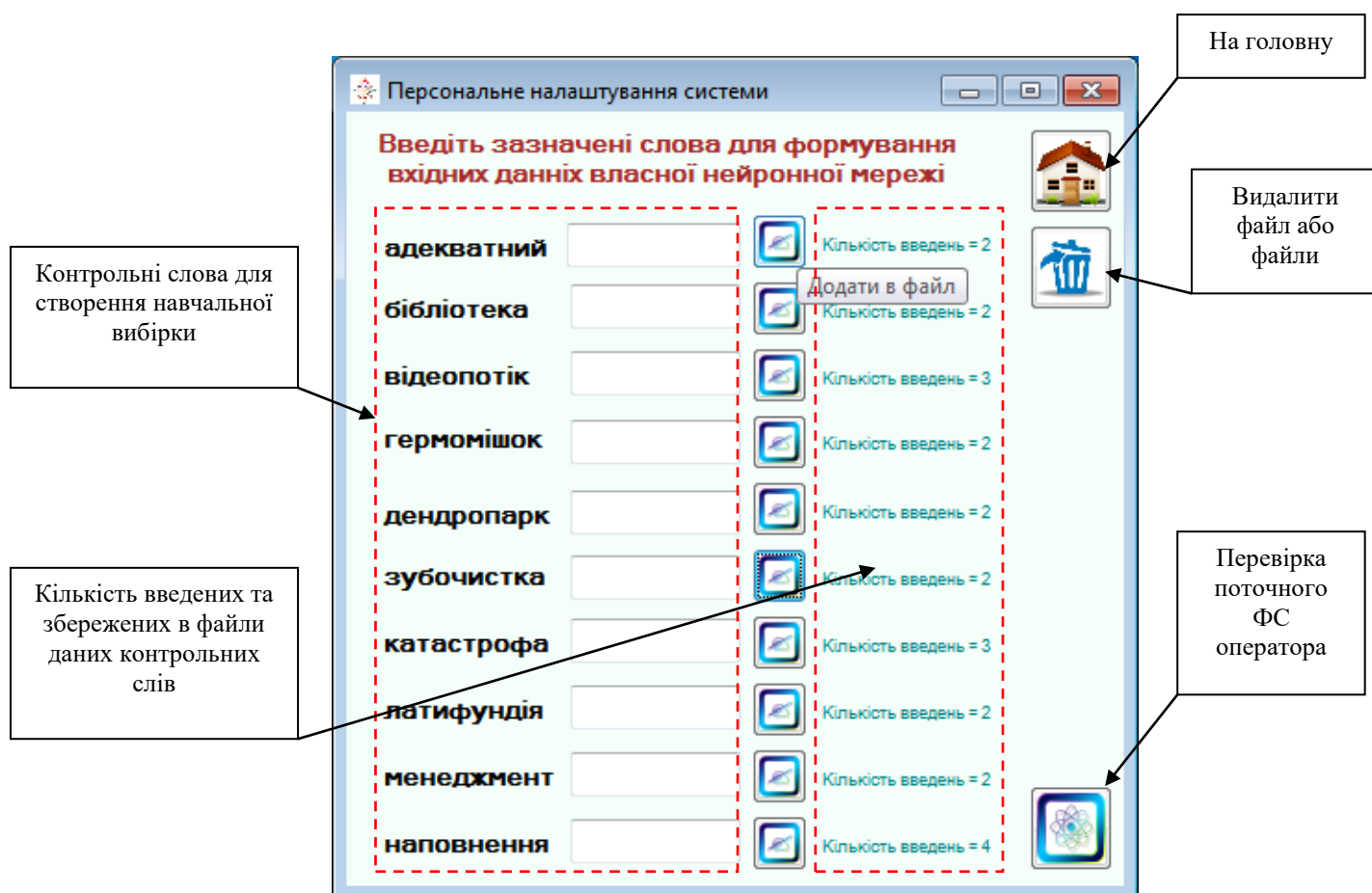


Рисунок Г.14 – Вікно «Персональні налаштування системи»

!!! Користувач обов'язково має заповнити файли для навчання нейронної мережі хоча б один раз, інакше НЕ зможе перевіряти власний функціональний стан. Для підказки використана технологія інформаційних повідомлень або повідомлень про помилки.

Відкривши вікно видалення файлів (рис. Г.15) користувач має обрати, який саме файл він бажає оновити, у випадку якщо це один файл, якщо всі то натиснути на кнопку видалення всіх файлів.

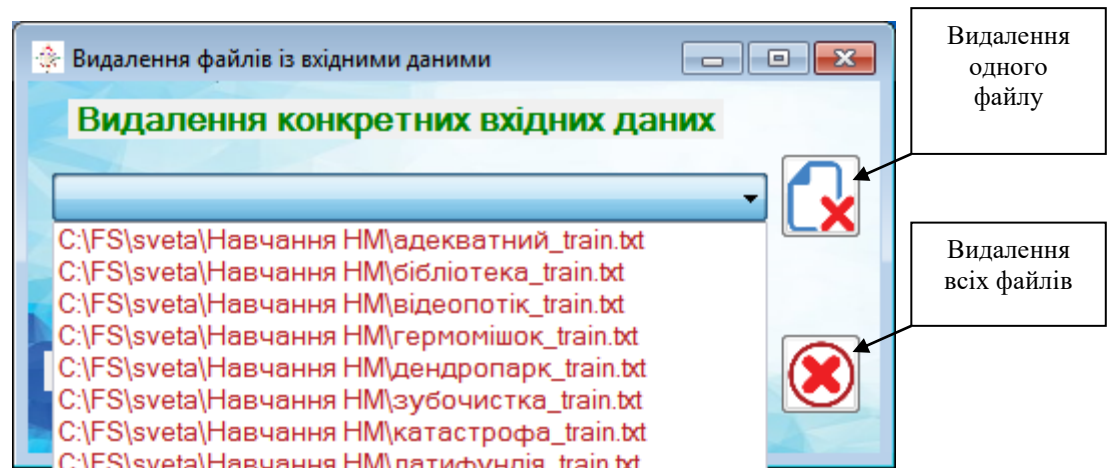


Рисунок Г.15 – Вікно видалення файлів

Натиснувши на «Перевірити ФС» користувач опиняється на вікні «Перевірка поточного ФС» (рис. Г.16). На цьому етапі роботи програми відбувається створення, навчання та симуляція Самоорганізованої мапи Кохонена, за якою визначається поточний стан оператора інформаційних систем. У результаті оператор отримує результат роботи мережі в текстовому файлі та у вигляді повідомлення (рис. Г.17).

Також після проходження перевірки можна перевірити статистичні дані, натиснувши кнопку «Перегляд статистики». У результаті відкриється Excel-файл (рис. Г.18) із такою інформацією як час проходження перевірки, її результат та за яким із слів вона відбувалася.

!!! Файл «Статистика» доступний лише для читання, тому вносити якісь нові дані неможливо.

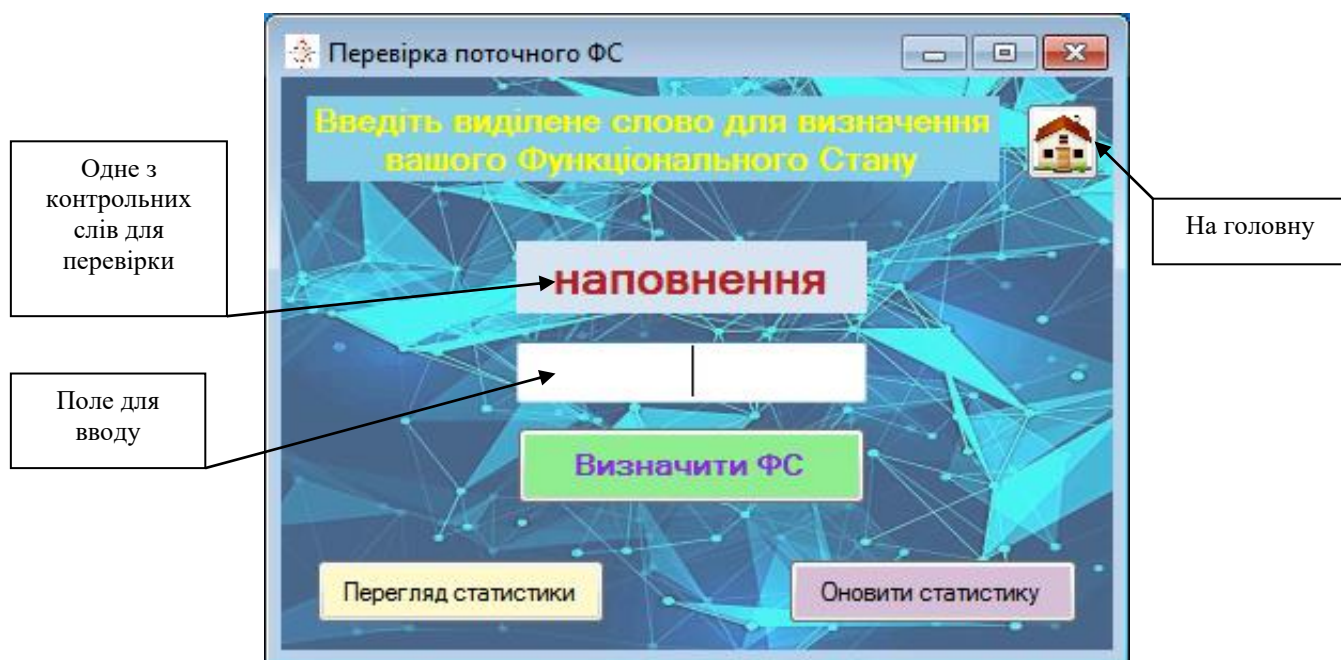


Рисунок Г.16 – Вікно «Перевірка поточного ФС»

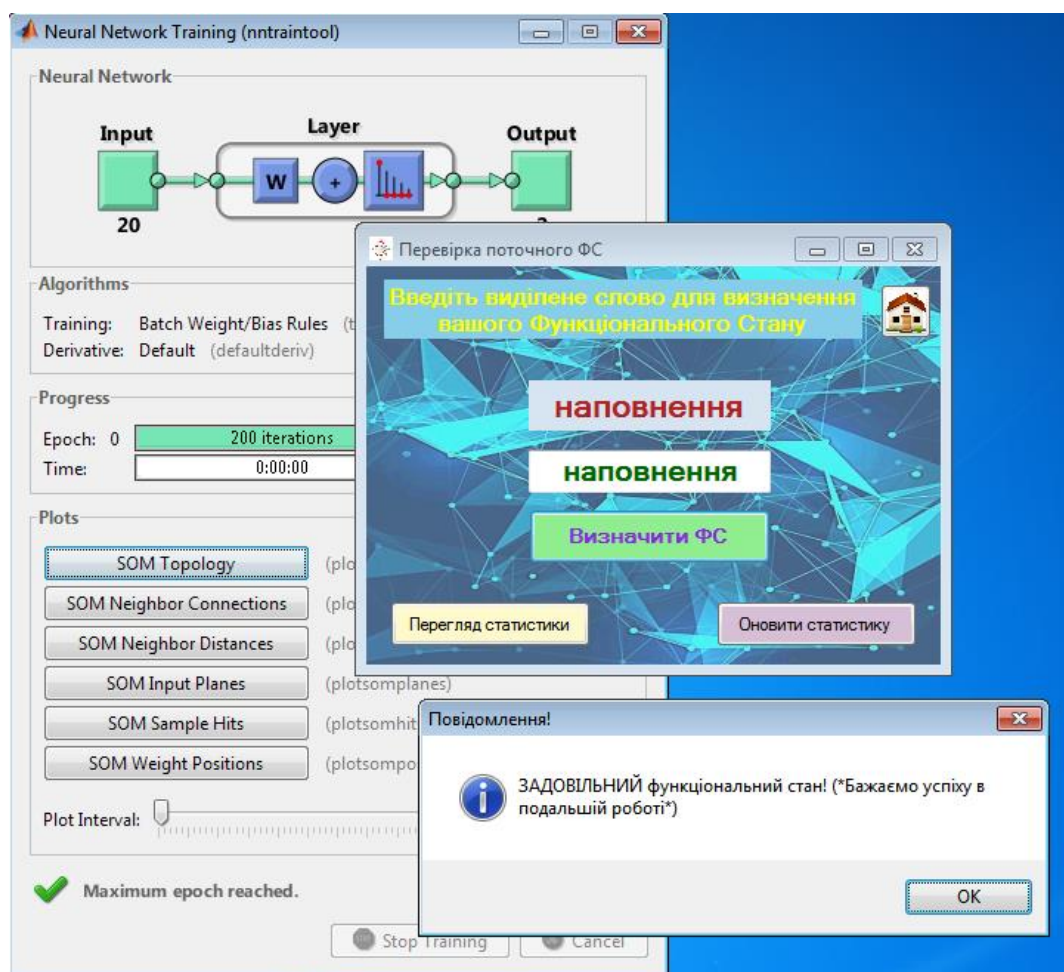
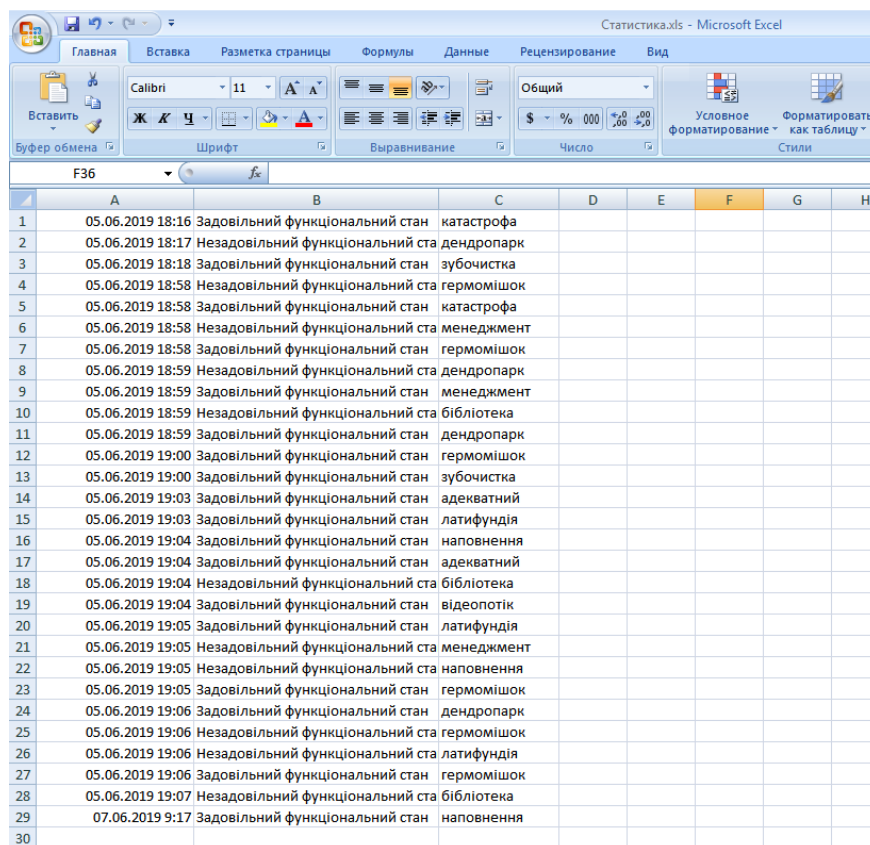


Рисунок Г.17 – Результат оцінки функціонального стану оператора



	A	B	C	D	E	F	G	H
1	05.06.2019 18:16	Задовільний функціональний стан	катастрофа					
2	05.06.2019 18:17	Незадовільний функціональний ста	дендропарк					
3	05.06.2019 18:18	Задовільний функціональний стан	зубочистка					
4	05.06.2019 18:58	Незадовільний функціональний ста	гермомішок					
5	05.06.2019 18:58	Задовільний функціональний стан	катастрофа					
6	05.06.2019 18:58	Незадовільний функціональний ста	менеджмент					
7	05.06.2019 18:58	Задовільний функціональний стан	гермомішок					
8	05.06.2019 18:59	Незадовільний функціональний ста	дендропарк					
9	05.06.2019 18:59	Задовільний функціональний стан	менеджмент					
10	05.06.2019 18:59	Незадовільний функціональний ста	бібліотека					
11	05.06.2019 18:59	Задовільний функціональний стан	дендропарк					
12	05.06.2019 19:00	Задовільний функціональний стан	гермомішок					
13	05.06.2019 19:00	Задовільний функціональний стан	зубочистка					
14	05.06.2019 19:03	Задовільний функціональний стан	адекватний					
15	05.06.2019 19:03	Задовільний функціональний стан	латифундія					
16	05.06.2019 19:04	Задовільний функціональний стан	наповнення					
17	05.06.2019 19:04	Задовільний функціональний стан	адекватний					
18	05.06.2019 19:04	Незадовільний функціональний ста	бібліотека					
19	05.06.2019 19:04	Задовільний функціональний стан	відеопотік					
20	05.06.2019 19:05	Задовільний функціональний стан	латифундія					
21	05.06.2019 19:05	Незадовільний функціональний ста	менеджмент					
22	05.06.2019 19:05	Незадовільний функціональний ста	наповнення					
23	05.06.2019 19:05	Задовільний функціональний стан	гермомішок					
24	05.06.2019 19:06	Задовільний функціональний стан	дендропарк					
25	05.06.2019 19:06	Незадовільний функціональний ста	гермомішок					
26	05.06.2019 19:06	Незадовільний функціональний ста	латифундія					
27	05.06.2019 19:06	Задовільний функціональний стан	гермомішок					
28	05.06.2019 19:07	Незадовільний функціональний ста	бібліотека					
29	07.06.2019 9:17	Задовільний функціональний стан	наповнення					
30								

Рисунок Г.18 – Вміст файлу «Статистика» одного з операторів

За допомогою кнопки «Оновити статистику» відбувається видалення файлу зі статистичними даними (рис. Г.19). Для створення нового слід пройти перевірку поточного функціонального стану, адже створення файлу автоматичне.

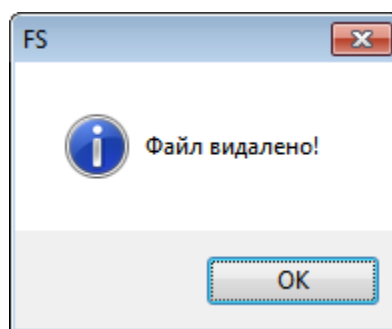


Рисунок Г.19 – Повідомлення про видалення файлу «Статистика»

Після кожного з переходів до нових вікон користувач має змогу повернутися на попереднє вікно за допомогою закриття поточного вікна.

У програмі передбачено велику кількість інформаційних повідомлень та повідомлень про помилки. Деякі з них наведені на рис. Г.20.

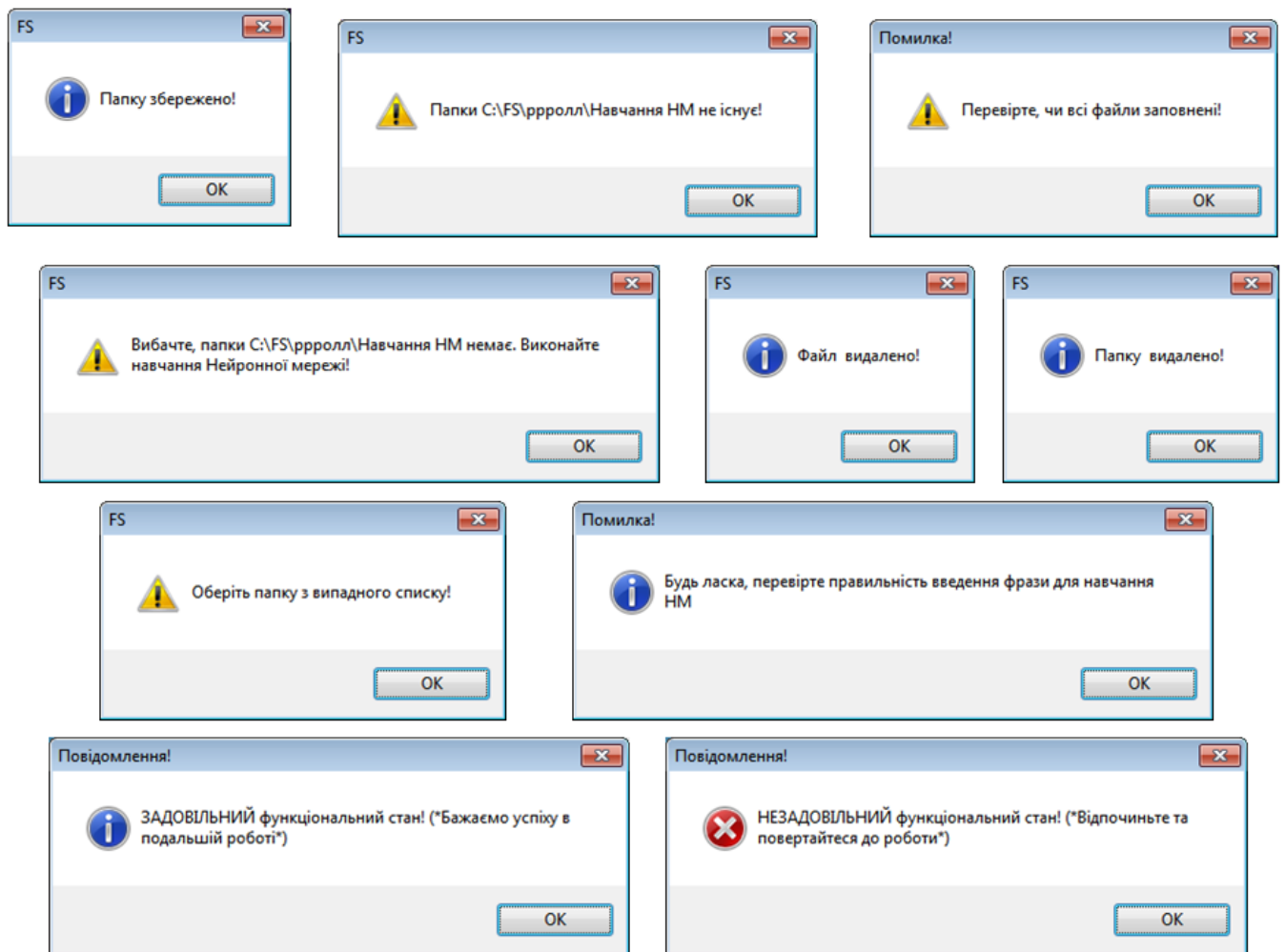


Рисунок Г.20 – Приклади деяких повідомлень, які видає програма «Визначення ФС»

Видалення інформаційної системи з комп'ютера

Один із найпростіших методів видалення програми – видалення за допомогою пакету установки встановленої програми. Для цього необхідно натиснути двічі лівою клав'яшею мишки на ярлик установки програми «Визначення ФС» – «Setup_FS». У результаті користувачу буде запропоновано або «Восстановить “Setup_FS”» або «Удалить “Setup_FS”» (рис. Г.21).

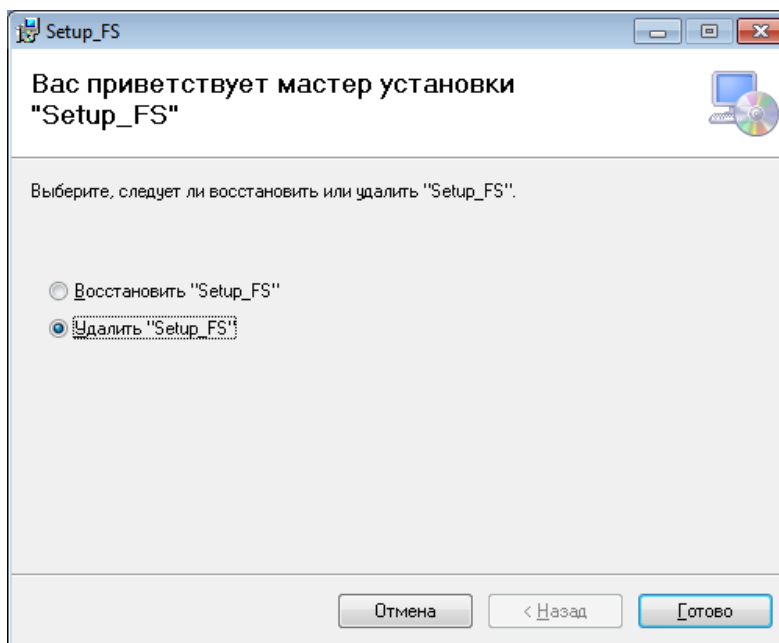


Рисунок Г.21 – Видалення програми «Визначення ФС» за допомогою майстра установки

Слід обрати «Удалить “Setup_FS”» для того, щоб програма «Визначення ФС» видалилася з комп’ютера оператора. У результаті натискання клавіші «Готово», вікно майстра установки видасть результат видалення (рис. Г.22).

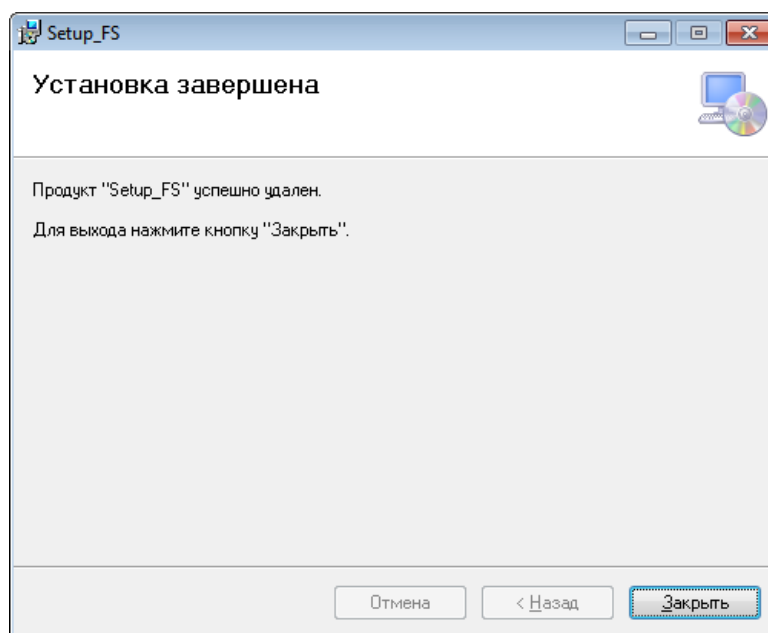


Рисунок Г.22 – Систему оцінки функціонального стану оператора видалено

ДОДАТОК Д. Копія акта впровадження



Карпуша В.Д.

«_____» _____ 2019 р.

Акт
Впровадження в навчальний процес
СУМСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ
результатів наукової роботи
студентки групи ІТ-51 Сумського державного університету
Вакал Світлана Миколаївна
на тему
«Автоматизація визначення функціонального стану операторів автоматизованих
технологічних комплексів»

Складений 5 січня 2019 р. комісією у складі:

Голова комісії:

Доцент кафедри комп'ютерних наук, зав. секції «Інформаційні технології проектування», кандидат технічних наук, доцент Шендрік В.В.

Члени комісії:

1. Професор кафедри комп'ютерних наук, доктор технічних наук, професор *Лавров Є.А.*
2. Доцент кафедри комп'ютерних наук, кандидат технічних наук, доцент *Чибіряк Я.І.*
3. Старший викладач кафедри комп'ютерних наук, кандидат технічних наук, *Кузнецов Е.Г.*

В період з 3 січня 2019 р. по 5 січня 2019 р. комісія провела роботу з визначення впровадження результатів Вакал С.М. в навчальний процес кафедри комп'ютерних наук.

Результати роботи комісії

1. На кафедру комп'ютерних наук передано комплекс програм «Автоматизація визначення функціонального стану операторів автоматизованих технологічних комплексів».
2. Матеріали використні в дисциплінах:
 - «Ергономіка» для слухачів магістратури, що навчаються за спеціальністю «Інформатика», при розробці теми «Атестація робочого місця» (лабораторна робота – 2 год.).
 - «Організація людино-машинної взаємодії» для слухачів магістратури, що навчаються за спеціальністю «Інформаційні технології проектування», при розробці теми «Ергономіка автоматизованих виробництв» (лабораторна робота – 2 год.).

Голова комісії
Члени комісії

ДОДАТОК Ж. Копії публікацій за темою роботи

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**



**ІНФОРМАТИКА,
МАТЕМАТИКА,
АВТОМАТИКА**

ІМА :: 2018

**МАТЕРІАЛИ
та програма**

**НАУКОВО-ТЕХНІЧНОЇ
КОНФЕРЕНЦІЇ**

(Суми, 05-09 лютого 2018 року)

**Суми,
Сумський державний університет
2018**

Комп'ютерна технологія прогнозування якості навчальної діяльності в модульних системах електронного навчання

Рудакова Н.О., студент; Вакал С.М., студент; Лавров Е.А., професор
Сумський державний університет, м. Суми
Барченко Н.Л., старший викладач
Сумський національний аграрний університет, м. Суми

Вступ. Вимогою до навчальних систем є здатність адаптації до індивідуальних особливостей кожної людини, що навчається, низький рівень якої призводить до відсутності мотивації у студентів та викладачів, що в свою чергу знижує ефективність електронного навчання.

Використання технології інтелектуального агента в навчальному середовищі забезпечить можливість вирішити задачу оптимізації якості людино-машинної взаємодії та дозволить враховувати мотивацію, рівень підготовки та інші особливості студентів. Необхідними є модуль прогнозування вхідних даних для оптимізації та модуль вибору раціональної технології навчання.

Постановка задачі. Розробити інформаційну технологію для вибору оптимальної стратегії, яку інтелектуальний агент повинен запропонувати людині-оператору. Цільовою функцією задачі оптимізації є максимально можлива ймовірність безпомилкового виконання, а обмеженнями доцільно вважати час виконання та рівень складності.

Задача оптимізації зведена до задачі лінійного програмування, в якій керованими змінними є бінарні змінні, що описують вибір можливих діалогових процедур.

Результати. Розроблена інформаційна технологія прогнозує ймовірність безпомилкового виконання модулю, математичне сподівання часу виконання та наявність відмови від діяльності, враховуючи індивідуальні параметри студента та зовнішнього середовища; вирішує задачу оптимізації, змінюючи функціональну мережу на кожному кроці управління діалогом в залежності від фактичних даних; пропонує користувачу раціональні технології навчання.

Висновки. Створені оптимізаційні моделі для визначення оптимальної стратегії людино-машинної взаємодії дозволяють підвищити ефективність навчання в системі "студент-комп'ютер" та забезпечити високий рівень інтерактивності і адаптивності електронного навчання.

Министерство образования и науки РФ ■ Петроавд-
ский государственный университет ■ Московский меж-
дународный университет ■ ООО «ФОРС – Центр разра-
ботки» ■ ООО «Интернет-бизнес-системы» ■ Институт
прикладных математических исследований КарИЦ РАН

ЦИФРОВЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ, НАУКЕ, ОБЩЕСТВЕ

Материалы XII всероссийской
научно-практической конференции

(4–6 декабря 2018 года)

Петроавдск
2018

- propelled by local industry in Russia // *ACM Inroads*. 2015. Vol. 6. Issue. 4. PP. 41–51.
3. Корзун Д. Ж., Систова Н. Ю., Боговиленский Ю. А., Борзин А. В. Применение математического образования при подготовке специалистов по разработке программного обеспечения в сфере информационно-коммуникационных технологий // *Материалы 16-й Открытой Всероссийск. конф. «Преподавание информационных технологий в Российской Федерации»*. М: МГТУ им. Н. Э. Баумана, 2018. С. 76–78.
 4. Воронин А. В., Боговиленский Ю. А., Корзун Д. Ж., Шабанов А. И. Обучение технологии разработки программного обеспечения в Петропавловском государственном университете // *Сб. докл. 5-й открытой всероссийской конф. «Преподавание информационных технологий в Российской Федерации»*. М.: АПКИТ, 2007. С. 102-119.
 5. Боговиленский Ю. А., Воронин А. В., Корзун Д. Ж. Организация годового курса «Технология разработки программного обеспечения» // *Материалы Всероссийск. науч.-практ. конф. «Информационная среда пути XXI века»*. Петропавловск. — 2007. — С. 37–39.
 6. Боговиленский Ю. А. Подготовка специалистов по информационным и коммуникационным технологиям на базе семейства стандартов «Прикладная математика и информатика» // *Материалы международной конференции «Развитие вычислительной техники в России и странах бывшего СССР: история и перспективы»*. Петропавловск: Изд-во ПетрГУ, 2006. Ч. 1. С. 33–45.
 7. Боговиленский Ю. А., Систова Н. Ю. Цифровая среда Института математики и информационных технологий: Распределенные информационно-образовательные ресурсы. 2018. В настоящем сборнике.
 8. Пономарев В. А., Боговиленский Ю. А. Цифровая среда Института математики и информационных технологий: Аппаратно-системная платформа. 2018. В настоящем сборнике.

ОБЕСПЕЧЕНИЕ КОГНИТИВНОГО КОМФОРТА В E-LEARNING

Е. А. Лавров, Н. Л. Барченко, С. М. Якал

Сумский государственный университет

Сумы

prof_lavrov@mail.ru

Рассмотрены проблемы создания адаптивных e-learning. Разработана информационная технология агента-менеджера. Приведены примеры.

Ключевые слова: эргономика, электронное обучение, когнитивный комфорт, адаптация

SUPPORT FOR COGNITIVE COMFORT IN E-LEARNING

E. A. Lavrov, N. L. Barchenko, S. M. Yakal

Sumy state university

Sumy

The problems of creating adaptive e-learning. Developed information technology for agent-manager. Examples are given.

Key words: ergonomics, e-learning, cognitive comfort, adaptation.

Исходные предпосылки. Массовое внедрение электронного обучения обострила проблема адаптации [1,9] системы к студенту и обеспечения когнитивного комфорта. Технология обучения должна определяться особенностями обучаемого и иметь возможность «приспосабливаться к нему». Очень часто студент бросает обучение т. к. «получает от системы не то, что ожидает» [1]. Это относится и к локальным системам, и, особенно, к системам дистанционного обучения.

Постановка задачи. В общем виде задача формулируется следующим образом. Имеется хранилище электронных учебных модулей. Модули могут различаться по стилю представления информации, сложности учебного материала и др. На первом этапе организации диалогового взаимодействия необходимо предоставить студенту учебный модуль, параметры которого наиболее полно соответствуют предпочтениям и целям студента. На втором этапе необходимо определить оптимальную стратегию человеко-машинного взаимодействия.

Результаты

Принципы построения моделей пользователей и систем e-learning. Решение задачи приспособления компьютера к студенту становится возможным в условиях АСУ вуза [1–3]. Именно тогда можно вести базу данных студентов и электронных средств обучения. Задача состоит в отнесении студента к одной группе. Причем каждой группе соответствует определенное множество значений характеристик обучаемого (психофизиологические характеристики, стили обучения и др.). Кроме того, заранее определяются возможные подходы к подаче учебного материала (с преобладанием текста, с преобладанием графики, с использованием аудиосредств, «от общего к частному» «от частного к общему» и т. п.). Необходимо найти однозначное соответствие студента характеристикам диалоговой системы. Для этого строится ряд моделей, описывающих предметную область. Модель обучаемого и системы обучения описывают множество разрезов, актуальных для конкретной системы обучения. Такими разрезами могут быть: «Психофизиология», «Стили обучения», «Мотивация», «Подготовленность по дисциплине», «Опыт работы с системой», «Функциональное состояние».

Функциональное состояние (ФС) студента может быть оценено с использованием различных аппаратных средств и методик [4], наиболее удобным подходом в наших условиях оказался метод анализа клавиатурного почерка Абашина В.Г. [5–6].

Система и альтернативные электронные модули, имеющиеся в базе данных e-learning должна быть описана в разрезах типа: Принятые способы адаптации, Формы подачи информации, Технологии навигации, Технологии и характеристики самоконтроля, Формы диалогового взаимодействия.

Управление процессом диалога и механизмы адаптации. Реализация интеллектуального адаптивного обучения предусматривается с использованием технологии специального агента-менеджера [7]. Задача состоит в возможности оперативной идентификации студента и подстановки личных (или групповых) данных о надежности и времени выполнения студентом операций взаимодействия с компьютером. Такие данные определяются путем ведения специальной системы статистического анализа успешности учебной деятельности разными студентами в разных условиях среды. Описание учебной деятельности производится с помощью специальной модели типа «функциональная сеть» проф. Губинского А.И. [8]. Существует возможность по-разному построить algo-

ритм обучения (взаимодействие студента с компьютером). Для оперативного получения актуальных значений параметров модели нами разработан специальный математический аппарат нейронно-функциональных сетей и набор специальных расчетных моделей, а также соответствующее программное обеспечение.

Основные задачи агента-менеджера:

1. Эргономическая экспертиза электронных учебных ресурсов (оценка и паспортизация);
2. Ведение модели студента (определение психофизиологических особенностей и предпочтений);
3. Оптимизация диалогового взаимодействия в системе обучения (выбор наилучшей по показателю когнитивного комфорта базовой платформы обучения (электронного модуля) и последующее построение индивидуального учебного маршрута)[9].

Программная реализация технологии. Разработанный комплекс предназначен для работы студента, преподавателя и эксперта-эргономиста. Представленный программный комплекс может работать с любой обучающей системой. В данном случае он интегрирован с системой Moodle (рис. 1).

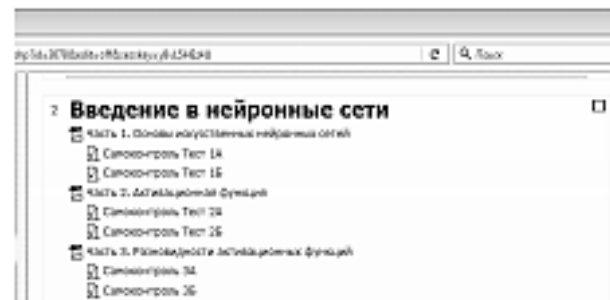


Рис. 1 Разработанный электронный курс в системе Moodle

В разработанном электронном курсе имеются в наличии различные способы самоконтроля (самоконтроль А и самоконтроль с последующим повторением проблемных фрагментов Б). Агент-менеджер позволяет построить оптимальную траекторию обучения с учетом ограничения на уровень сложности учебного материала и времени, которое имеется в наличии.

Пример рекомендаций по индивидуальной траектории обучения представлен на рис. 2.



Рис. 2 Пример индивидуальной траектории обучения

Выводы. Предложенная оптимизационная модель выбора варианта человеко-машинного взаимодействия в системе электронного обучения, обеспечивающая максимум когнитивного комфорта и максимум итоговой балльной оценки, позволяет оценить варианты взаимодействия и выбрать оптимальный вариант.

Библиографический список

1. Лавров Е. А. Подход к обеспечению эргономического качества информационной среды вуза // Труды Международной научно-практической конференции «Психология труда, инженерная психология и эргономика 2014» (Эрго 2014) Под редакцией А. Н. Анохина, П. И. Падерно, С. Ф. Сергеева. Санкт-Петербург, 2014. — С. 70–76.
2. Лавров Е.А., Клименко А. В. Компьютеризация управления вузом. — Сумы: Издательство «Довкілля», 2005. — 307с.
3. Лавров Е.А., Клименко А.В., Палт М.В., Трубищков Ю. В. Система компьютерного управления университетом. — М: Экономический факультет МГУ им. Ломоносова, ТЕИС, 2005. — 32с
4. Лавров Е.А., Барченко Н. Л. Измерение параметров оператора для систем эргономического обеспечения обучающихся сред // Вісник Сумського національного аграрного ун-ту.-Сер.«Механізація та автоматизація виробничих процесів». — Сумы, 2011. — Вып.8(23). — С.110–121
5. Абашин В. Г. Адаптивная математическая модель мультиметрической подсистемы определения работоспособности человека-оператора АРМ на основе нечетких множеств // Информационные системы и технологии. 2011. — № 5(67). С. 90–96
6. Лавров Е.А., Барченко Н. Л., Демиденко Д. Перспективы оценки функционального состояния оператора в системе «студент-компьютер» // Матеріали Третьої міжнародної науково-практичної конференції, м. Суми, 14–16 травня 2014 р. — Сумы: СумДУ, 2014 — С. 93–94
7. Лавров Е.А., Барченко Н. Л. Агент-менеджер в системе эргономического обеспечения электронного обучения // Бионика интеллекта. — 2013. — № 2 (81). — С. 115–120
8. Информационно-управляющие человеко-машинные системы: Исследование, проектирование, испытания: Справочник/ Адаменко А. Н., Ашеров А. Т., Лавров Е. А. и др. под общ. ред. Губинского А. И. и Енграфова Е. Г.- М., Машиностроение, 1993. — 528 с.
9. Лавров Е.А., Барченко Н. Л. Многоуровневая адаптация в университетских обучающих средах// В сборнике: Научно-образовательная информационная среда XXI века. Материалы IX Всероссийской научно-практической конференции, Петрозаводск, Н. С. Рузанова (отв. редактор). 2015.- Петрозаводск, 2015.- С. 118–122.



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ



ІНФОРМАТИКА, МАТЕМАТИКА, АВТОМАТИКА

ІМА - 2019

**МАТЕРІАЛИ
та програма**

**НАУКОВО-ТЕХНІЧНОЇ
КОНФЕРЕНЦІЇ**

(Суми, 23-26 квітня 2019 року)

Суми,
Сумський державний університет
2019

Інформаційна технологія оцінки функціонального стану операторів інформаційних систем

Вакал С.М., студент; Лавров Є.А., професор
Сумський державний університет, м. Суми, Україна

Вступ. Серед сьогоденної комп'ютеризації та інформатизації особлива роль відводиться задачам захисту інформації та необмеженості доступу до управління інформаційних систем та до їх ресурсів, захисту від несанкціонованого доступу.

Методи аутентифікації за біометричними параметрами особистості здатні забезпечити підвищену точність. До таких характеристик, що описує підсвідомі дії звичні для студента, відноситься і клавіатурний почерк. Він характеризує динаміку введення деякої фрази за допомогою клавіатури.

Постановка задачі. Створити гібридну систему управління доступом, що поєднує в собі як унікальні знання, так і біометричну характеристику студента (клавіатурний почерк) та дослідити особливості аутентифікації за почерком студента при введенні ним парольних даних.

Результати. Реалізовано програму, яка візуально оцінює вихідні параметри клавіатурного почерку окремого студента (після введення паролю студент може переглянути результати його введення: інтервали між натисканням та удержуванням клавіш, їх число рівне кількості символів введеного повідомлення; у полі інтервалу між клавішами значення завжди рівне нулю); розраховує статистику за серією введів (після багаторазового збереження введених даних є можливість обробки даних із застосуванням методів математичної статистики – розрахунок математичного очікування, дисперсії, середньоквадратичного відхилення, матриці кореляції між стовбцями та коваріаційна матриця; дані результати зберігаються в текстових файлах).

Програма функціонує в двох режимах: налагодження та аналіз. У результаті програми перевіряється вектор біометричних параметрів студента з еталонним вектором і виводить повідомлення чи дійсно цей вхід здійснює студент, що пройшов режим налагодження.

Висновки. Розроблено програму по аутентифікації студента за клавіатурним почерком при введенні парольних даних, яка здатна розрізнати людей, за допомогою біометричних параметрів.

**ДОДАТОК 3. Копії дипломів переможця конкурсів студентських
наукових робіт**



Міністерство освіти і науки України
Вінницький національний технічний університет

Дипломом переможця II ступеня нагороджується

ВАКАЛ Світлана Миколаївна

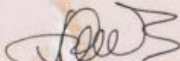
Сумський державний університет



Всеукраїнський конкурс
студентських наукових робіт з напрямку
«Інформатика і кібернетика»

Вінниця

25 -26 квітня 2019 р.

Голова конкурсної комісії,
проректор з наукової роботи ВНТУ, д.т.н., проф.  С. В. Павлов