

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ МОДЕЛЮВАННЯ

(назва теми)

РОЗУМОВОЇ ДІЯЛЬНОСТІ ЛЮДИНИ. SOFTWARE TO

MODEL HUMAN MENTAL ACTIVITY

Виконав(ла): студент(ка) 4 курсу, групи КНТ-129

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

КАРНАУХ В. Д.

(ПРІЗВИЩЕ та ініціали)

Керівник СЕРДЮК С. М.

(ПРІЗВИЩЕ та ініціали)

Рецензент ПОЛЯКОВ М. О.

(ПРІЗВИЩЕ та ініціали)

2023__

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

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

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА(КИ)

КАРНАУХ Вероніки Дмитрівни
(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проекту (роботи) Програмне забезпечення для моделювання розумової діяльності людини. Software to Model Human Mental Activity

керівник проекту (роботи) к.т.н., доцент, СЕРДЮК Сергій Микитович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від « 4 » квітня 2023 року № 87

2. Строк подання студентом проекту (роботи) 1 червня 2023 року

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

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Розробка структури програмного забезпечення. 3. Основні рішення щодо реалізації роботи системи. 4. Керівництво програміста. 5. Керівництво оператора. Додатки.

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

Слайди презентації

6. Консультанти розділів проекту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання видав
1-5 Основна частина	СЕРДЮК С.М., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання « 03 » березня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2 тиждень	Розділ 1
3	Розробка архітектури програми.	3 тиждень	Розділ 2
4	Розробка програми.	4–5 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми.	6 тиждень	Розділ 4
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	7 тиждень	Розділ 5, Додатки
7	Захист роботи.	8 тиждень	

Студент(ка)

_____ Вероніка КАРНАУХ
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проекту (роботи)

_____ Сергій СЕРДЮК
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до випускної кваліфікаційної роботи бакалавра: 93 с., 8 табл., 15 рис., 2 дод., 11 джерел.

Р-АДИЧНА СИСТЕМА, АЛГОРИТМ, ІДЕЯ, МОЗОК, ПРОЦЕСИ МИСЛЕННЯ ЛЮДИНИ, ПІДСВІДОМІСТЬ, СВІДОМІСТЬ.

Об'єкт дослідження – процес мислення людини.

Предмет дослідження – програмне забезпечення для моделювання розумової діяльності людини.

Мета роботи – розробка програмного забезпечення, що дозволить моделювати мислячу систему людини на основі р-адичних дерев.

Матеріали, методи та технічні засоби: мова програмування C++; середовище розробки Qt Creator 5.0.1; об'єктно-орієнтоване програмування; персональний комп'ютер із процесором AMD Athlon II X4 635 під управлінням операційної системи Microsoft Windows 7.

Результати. Проаналізовано роботу розумової діяльності людини за обраною моделлю мислення та реалізовано її у вигляді програмного забезпечення для подальших наукових досліджень.

Висновки. Розроблено програмне забезпечення, що відображає роботу обраної моделі мислення, надає можливість користувачу формально описувати інформацію, що надходить на аналізатори людини, та досліджувати її реакції та вибори на основі індивідуальних характеристик суб'єктів.

Галузь використання – штучний інтелект.

ABSTRACT

Explanatory note to the final qualifying work of the bachelor: 93 pages, 8 figures, 15 tables, 2 appendixes, 11 sources.

P-ADIC SYSTEM, ALGORITHM, IDEA, BRAIN, HUMAN PROCESS THINKING, SUBCONSCIOUSNESS, CONSCIOUSNESS.

Object of study is the process of human thinking.

Subject of study is the computer software for human mental activity modeling.

The purpose of the work is to create a software system that will allow modeling the human thinking system based on p-adic trees.

Materials, methods, and tools: programming language C++; integrated development environment Qt Creator 5.0.1; object-oriented programming; personal computer with an AMD Athlon II X4 635 processor under the control of the Microsoft Windows 7 operating system.

Results. Human thinking process based on a chosen model of thinking is analyzed and is implemented as a software system for the future scientific studies.

Conclusions. A software system that reflects the work of the described model of thinking, allows user to formally define information received by human analyzers and explore their reactions and choices based on the individual characteristics of subjects has been developed.

The field of use is a artificial intelligence.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок.....	8
Вступ	9
1 Аналіз предметної області	10
1.1 Огляд програмних аналогів	10
1.1.1 Чат-бот ChatGPT	10
1.1.2 Платформа Sophia 2020	11
1.2 Технічне завдання	13
1.2.1 Підстави для розробки	13
1.2.2 Призначення розробки	13
1.2.3 Вимоги до програми	13
1.2.3.1 Вимоги до функціональних характеристик.....	13
1.2.3.2 Вимоги до надійності.....	14
1.2.3.3 Умови експлуатації	14
1.2.3.4 Вимоги до складу та параметрів технічних засобів	14
1.2.3.5 Вимоги до інформаційної та програмної сумісності.....	15
1.2.3.6 Вимоги до програмної документації.....	15
1.2.4 Стадії та етапи розробки	15
1.2.5 Порядок приймання і тестування.....	15
2 Розробка структури програмного забезпечення	16
2.1 Вибір інструментарію розробки програмного забезпечення.....	16
2.1.1 Огляд особливостей мови програмування	16
2.1.2 Огляд особливостей середовища розробки	16
2.2 Моделювання програмного забезпечення	17
2.2.1 Діаграма активностей	17
2.2.2 Діаграма прецедентів	19
2.2.3 Діаграма послідовностей	20
3 Основні рішення щодо реалізації компонентів системи	21
3.1 Розробка алгоритму.....	21

3.2 Основні рішення щодо розроблених класів	24
3.2.1 Перелічення	24
3.2.2 Клас Reactions	24
3.2.3 Клас Knowledge.....	25
3.2.4 Клас KnowledgeBase	25
3.2.5 Клас Human	26
3.2.6 Діаграма класів.....	28
3.3 Розробка графічного інтерфейсу	30
4 Керівництво програміста.....	32
4.1 Призначення та умови застосування програми.....	32
4.1.1 Призначення програми.....	32
4.1.2 Функції програми.....	32
4.1.3 Умови застосування програми	32
4.2 Характеристики програми	33
4.2.1 Структура програми	33
4.2.2 Алгоритм роботи програми	34
4.3 Звертання до програми	34
4.4 Початкові та вихідні дані.....	34
5 Керівництво оператора.....	35
5.1 Призначення програми	35
5.2 Умови виконання програми	35
5.2.1 Апаратні вимоги програми	35
5.2.2 Вимоги до користувача	35
5.3 Виконання програми	35
5.4 Приклад роботи програми	37
Висновки	43
Перелік джерел посилань	44
Додаток А Текст програми.....	46
Додаток Б Слайди презентації	85

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

$D_{\text{під},f}$	– база заборонених ідей у КЦП;
$D_{\text{під},i}$	– база цікавих ідей у КЦП;
$D_{\text{св},f}$	– база заборонених ідей у КЦС;
$D_{\text{св},i}$	– база цікавих ідей у КЦС;
F_{max}	– поріг надзвичайної протидії ідеї у КЦС;
F_{mem}	– поріг протидії ідеї для зберігання у ДП;
$F(J)$	– міра протидії ідеї;
F_{+}	– поріг протидії ідеї для зберігання у ДП без зміни у часі;
I_{max}	– поріг надзвичайного інтересу ідеї у КЦС;
I_{mem}	– поріг інтересу ідеї для зберігання у ДП;
I_{min}	– поріг мінімального інтересу ідеї у КЦС;
$I(J)$	– міра інтересу ідеї;
I_{+}	– поріг інтересу ідеї для зберігання у ДП без зміни у часі;
J	– ідея-аттрактор;
O_d	– область сумнівів;
S_{mem}	– поріг сили ідеї для зберігання у ДП;
$S(J)$	– сила ідеї;
S_{+}	– поріг сили ідеї для зберігання у ДП без зміни у часі;
T_{i1}	– поріг узгодженості у КЦП;
T_{i2}	– поріг фізичної реалізації у КЦП;
T_m	– поріг реалізації у КЦС;
$T(J)$	– міра узгодженості ідеї;
ДП	– довгострокова пам'ять;
КЦП	– керуючий центр підсвідомості;
КЦС	– керуючий центр свідомості;
ОП	– оперативна пам'ять.

ВСТУП

Мислення є невід'ємною властивістю мозку людини, дослідження якої є задачею різноманітних організацій світу. Створення людиноподібного мислячого механізму (та/або алгоритму) може вважатися одним із найважливіших аспектів розуміння законів природи і досягнення вершини можливостей людства.

Але якщо говорити про найрозповсюдженіші цілі, то зазвичай виконується спроба створити ідеальну систему, що не буде генерувати помилки та працювати згідно до них, тоді ж як мислення живого індивіду може бути класифіковано як правильне або неправильне лише суб'єктивно та за певними нормами (моралі, закону, сподобань тощо). Можна зробити припущення, що саме можливість робити помилковий вибір відрізняє людину від роботів.

Створення програмного забезпечення, імітуючого роботу мозку людини, є актуальною темою для дослідження як можливостей сучасного програмування, так і деяких закономірностей процесів мислення.

Метою даної кваліфікаційної роботи бакалавра є розробка програмного забезпечення, що базується на математичній моделі мозку О. Хреннікова [1] та імітує роботу мислячої системи людини при надходженні інформації на зоровий аналізатор людини. За умови успішного застосування можливе подальше покращення та розширення розроблюваної системи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

У даному розділі було розглянуто існуючі методи моделювання та представлення процесів мислення людини, а також вимоги до розроблюваного програмного забезпечення, заснованого на моделі О.Ю. Хреннікова [1].

1.1 Огляд програмних аналогів

Мислення – це процес познання людиною навколишнього середовища, обробки доступної інформації та здійснення виборів відповідно до наявних умов і власного досвіду. Сучасні методи імітації роботи мозку людини не здатні відтворити його характеристики та особливості у повному обсязі, є вузьконаправленими і зазвичай обмеженими певним набором функціональних можливостей. Це ускладнює порівняння існуючих систем штучного інтелекту і залишає широке коло задач для дослідження та розробки.

Далі наведено деякі приклади моделювання мислення людини.

1.1.1 Чат-бот ChatGPT

ChatGPT – це чат-бот, заснований на нейромережевій моделі GPT (Generative Pre-trained Transformer), що генерує людиноподібні відповіді у процесі спілкуванні із користувачем, розроблений компанією OpenAI [2].

Основні можливості ChatGPT [2]:

- генерування відповідей на запитання та запити, надані користувачем у вигляді тексту та картинок;
- підтримка зв'язної розмови із користувачем, що імітує діалог із реальною людиною;
- виконання творчих завдань, таких як написання літературних творів (віршів, оповідань, п'єс), надання порад і створення описів об'єктів для різних проєктів;

- підтримка синтаксично та семантично коректних перекладів різними мовами;
- здатність до самонавчання на основі доступних наборів даних.

Одним із основних недоліків ChatGPT є можливість генерування неправильних відповідей або відповідей, що ідейно не стосуються запиту [3].

Приклад роботи ChatGPT наведено на рис. 1.1.

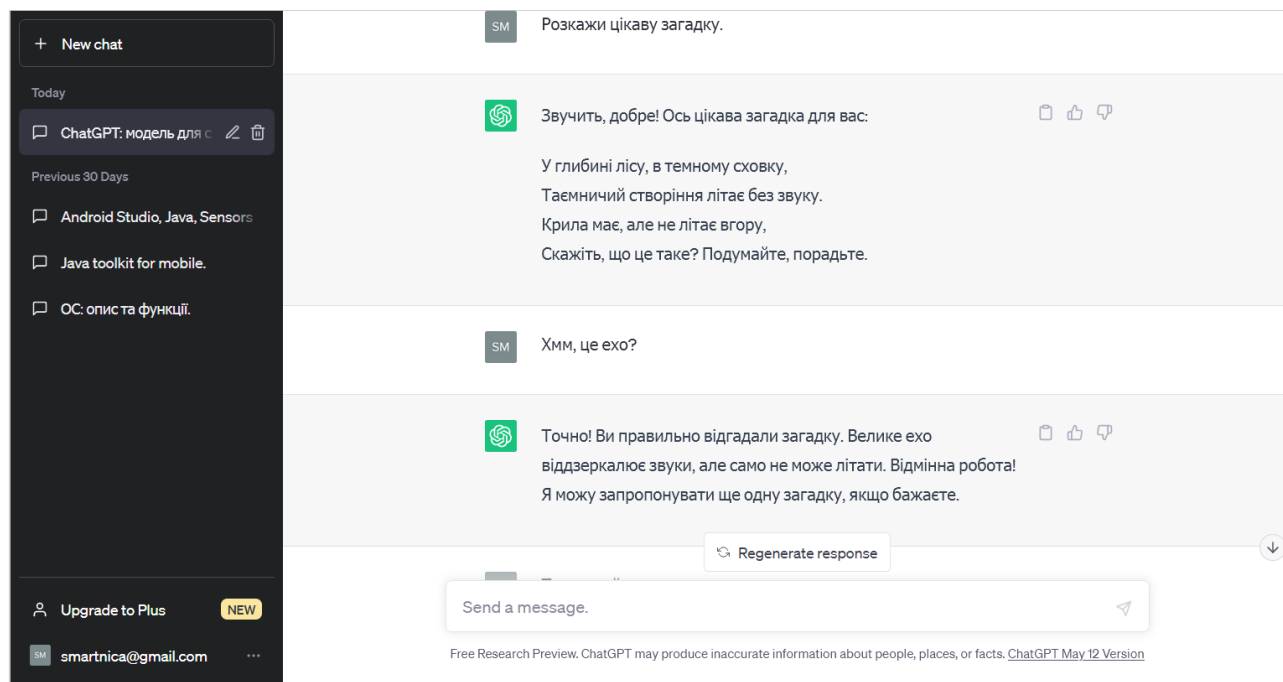


Рисунок 1.1 – Скріншот прикладу роботи ChatGPT [2]

1.1.2 Платформа Sophia 2020

Sophia 2020 – це платформа розробки (а також службовий робот), призначена для дослідження можливостей роботи штучного інтелекту при виконанні запрограмованих завдань і комунікації у людському просторі, що було створено Hanson Robotics [4].

Основні можливості Sophia 2020 [4]:

- імітація жестів і виразів обличчя людини;
- підтримка простої розмови на попередньо визначені теми за допомогою сенсорів і технології розпізнавання мовлення.

Одним із недоліків Sophia 2020 є неможливість генерування нових, непередбачених у коді рішень, тобто всі можливі реакції реакції та відповіді задаються програмно [4].

На рис. 1.2 наведено структурну схему платформи Sophia 2020.

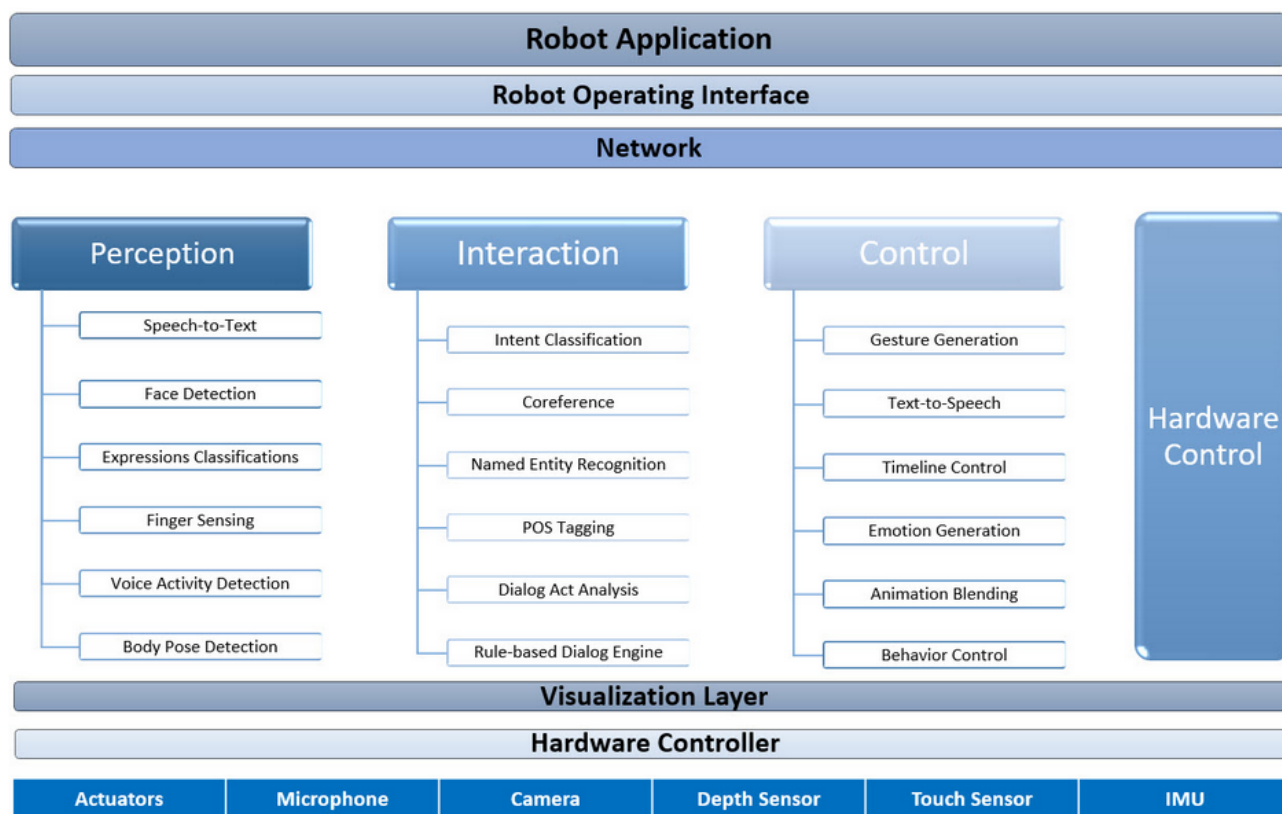


Рисунок 1.2 – Структурна схема Sophia 2020 [4]

Розглянуті вище системи не відповідають на питання "Чому було зроблено саме такий вибір у певній ситуації?" і не передбачають вибору між декількома варіантами, заснованому на індивідуальних параметрах і різному отриманому досвіду.

Таким чином, модель мислення, що розглядається Хренніковим [1], не може бути реалізована у перелічених системах через унікальність заданих умов. У результаті було сформульовано технічне завдання на програмне забезпечення, що розроблюється.

1.2 Технічне завдання

1.2.1 Підстави для розробки

Підставою для розробки програмного забезпечення служить завдання на дипломну кваліфікаційну роботу бакалавра на тему "Програмне забезпечення для моделювання розумової діяльності людини", затверджене наказом № 87 від 04 квітня 2023 р. по НУ "Запорізька політехніка".

1.2.2 Призначення розробки

Дане програмне забезпечення призначене для дослідження процесів мислення людини у залежності від заданих користувачем параметрів і заданих програмно досвіду і набору реакцій.

1.2.3 Вимоги до програми

1.2.3.1 Вимоги до функціональних характеристик

Розроблюване програмне забезпечення має надавати користувачу наступні можливості:

- вибір індивіду (із початково заданими психофізіологічними параметрами та досвідом);
- зміна окремих параметрів індивіду (певні параметри мозку; пороги мінімального інтересу ідеї, надзвичайних інтересу та протидії ідеї; пороги інтересу, протидії та сили ідеї для зберігання у довгостроковій пам'яті та зберігання у довгостроковій пам'яті без зміни у часі);
- встановлення параметрів ситуації (стать, колір волосся, вік, зріст, предмет, соціальний рівень);
- відтворення ситуації для обраного індивіду;
- перегляд результатів експериментів.

Вхідними даними, необхідними для функціонування програмного продукту, є параметри індивіду, бази даних цікавих і заборонених ідей центрів свідомості і підсвідомості, закодовані послідовності відтворюваних ситуацій. Вихідними даними є змінені бази цікавих і заборонених ідей центру свідомості, а також виведена на дошку інформація щодо проведеного експерименту.

1.2.3.2 Вимоги до надійності

Розроблюване програмне забезпечення повинно не допускати введення користувачем некоректних вхідних даних. Постійні індивідуальні параметри та зміни характеристик індивіду (наприклад, видалення ідеї із бази даних цікавих або заборонених ідей центру свідомості) має бути виведено на інтерфейс.

1.2.3.3 Умови експлуатації

Умови експлуатації повинні відповідати вимогам, які пред'являються до технічних засобів. Показники температури, відносної вологості і швидкості руху повітря повинні відповідати оптимальним і допустимим значенням в робочій зоні виробничих приміщень.

1.2.3.4 Вимоги до складу та параметрів технічних засобів

Серед технічних засобів мають бути наявні:

- ОС Windows 7 та вище;
- ОЗУ 512 МБ або більше;
- наявність вільного місця на жорсткому диску більше 512 МБ;
- наявність клавіатури, миші, монітору.

1.2.3.5 Вимоги до інформаційної та програмної сумісності

Розроблюване ПЗ повинно підтримувати спільний режим роботи з ОС Windows. Програма повинна бути підготовлена до можливої модернізації. Для реалізації програмного продукту необхідно використовувати мову програмування C++ та середовище розробки Qt Creator.

Інформаційні структури, використані на вході та на виході, мають відповідати визначеним структурам даних.

1.2.3.6 Вимоги до програмної документації

Включно із розроблюваним програмним забезпеченням повинно бути розроблено керівництва програміста та оператора, а також опис програми.

1.2.4 Стадії та етапи розробки

Розробка даного програмного забезпечення повинна включати наступні етапи:

- дослідження та аналіз теорії процесів мислення людини;
- проектування й опис архітектури;
- розробка програми;
- тестування програми.

1.2.5 Порядок приймання і тестування

Тестування даного програмного забезпечення проводиться у вигляді виконання усіх можливих сценаріїв оператора.

Дане програмне забезпечення вважається прийнятним, якщо воно задовольняє усім висунутим вимогам до нього і успішно пройшло етап тестування.

2 РОЗРОБКА СТРУКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У даному розділі було розглянуто особливості програмних засобів для реалізації дипломної кваліфікаційної роботи, а також основні рішення щодо логічної структури розроблюваної програми.

2.1 Вибір інструментарію розробки програмного забезпечення

2.1.1 Огляд особливостей мови програмування

C++ – це мова програмування високого рівня, чий синтаксис заснований на синтаксисі C, із підтримкою декількох парадигм програмування: об'єктно-орієнтованої, процедурної, узагальненої та функціональної. C++ має багату стандартну бібліотеку, що включає до себе поширені контейнери (такі як вектори та списки), алгоритми (наприклад, пошуку та циклів), методи введення-виведення та інші можливості [5].

У таблиці 2.1 наведено порівняння розповсюджених мов програмування із мовою, обраною для реалізації програмного забезпечення.

Таблиця 2.1 – Порівняння мов програмування [5]-[7]

	C	C++	Java	Python
Компільована/інтерпретована	К	К	К	І
Мультипарадигмальність	+	++	+++	+++
Безпечність типізації	-	-	+	+
Вивід типів	-	+-	-	+
Стандартизованість	+	+	-	-
Ефективність	+++	++	++	+

2.1.2 Огляд особливостей середовища розробки

Qt Creator – це крос-платформне інтегроване середовище розробки, що використовує бібліотеки фреймворку Qt. Це IDE підтримує популярні системи контролю версій (такі як Git, Subversion та інші), надає функції автозаповнення,

стилізації, підсвічування та рефакторингу коду, а також зручні візуальні засоби розробки інтерфейсу із використанням QtWidgets у вбудованому редакторі форм (Qt Designer) [8].

У таблиці 2.2 наведено порівняння популярних середовищ розробки для програмування мовою C++ із обраним IDE Qt Creator.

Таблиця 2.2 – Порівняння середовищ розробки [8]-[9]

	Microsoft Visual Studio	NetBeans	Qt Creator
Багатоплатформність	+-	+	+
Бібліотеки інтерфейсу	+	-	+
Статичний аналіз коду	+	-	+
Безкоштовність	+-	+	+
Підтримка репозиторіїв	++	+	++

2.2 Моделювання програмного забезпечення

На етапі проєктування було розроблено ряд моделей у вигляді UML-діаграм для відображення логічної структури програмного забезпечення. Більш детально алгоритм роботи програми та класову ієрархію розглянуто у розділі 3.

2.2.1 Діаграма активностей

Потік подій для розробленої діаграми активностей.

Крок 1. Користувач вводить дані щодо індивіду та ситуації до програмного інтерфейсу та підтверджує відтворення ситуації.

Крок 2. Дані перетворюються у код та передаються до алгоритму роботи.

Крок 3. Відбувається пошук у базах цікавих і заборонених ідей центру підсвідомості.

Крок 4. Якщо збіг знайдено, то перехід до кроку 5, інакше – до кроку 9.

Крок 5. Розраховуються $I(J)$, $F(J)$, $T(J)$ та порівнюються із T_{i1} і T_{i2} .

Крок 6. Якщо пороги пройдено, то перехід до кроку 7, інакше – до кроку 9.

Крок 7. Визначається реакція на ситуацію.

Крок 8. Перехід до кроку 23.

Крок 9. Відбувається пошук у базах цікавих і заборонених ідей центру свідомості.

Крок 10. Якщо збіг знайдено, то перехід до кроку 11, інакше – до кроку 18.

Крок 11. Розраховуються $I(J)$, $F(J)$, $T(J)$, $S(J)$ та порівнюються із $I_{\min}$, $I_{\max}$, $F_{\max}$, T_m .

Крок 12. Якщо ситуація однозначно цікава, то перехід до кроку 13, інакше – до кроку 15.

Крок 13. Визначається реакція на ситуацію.

Крок 14. Перехід до кроку 20.

Крок 15. Якщо ситуація сумнівна, то перехід до кроку 16, інакше – до кроку 18.

Крок 16. Визначається сумнівність ситуації.

Крок 17. Перехід до кроку 20.

Крок 18. Визначається відсутність реакції на ситуацію.

Крок 19. Перехід до кроку 20.

Крок 20. $I(J)$, $F(J)$, $S(J)$ порівнюються із I_{mem} , F_{mem} , S_{mem} , I_+ , F_+ , S_+ .

Крок 21. Якщо пороги пройдено, то перехід до кроку 22, інакше – до кроку 23.

Крок 22. Ідея додається до довгострокової пам'яті (бази цікавих ао заборонених ідей відповідно).

Крок 23. Передаються дані щодо реакції індивіду на ситуацію.

Крок 24. Отримані дані виводяться на інтерфейс програмного забезпечення.

Крок 25. Користувач отримує дані щодо результатів експерименту.

Діаграму активностей наведено на рис. 2.1.

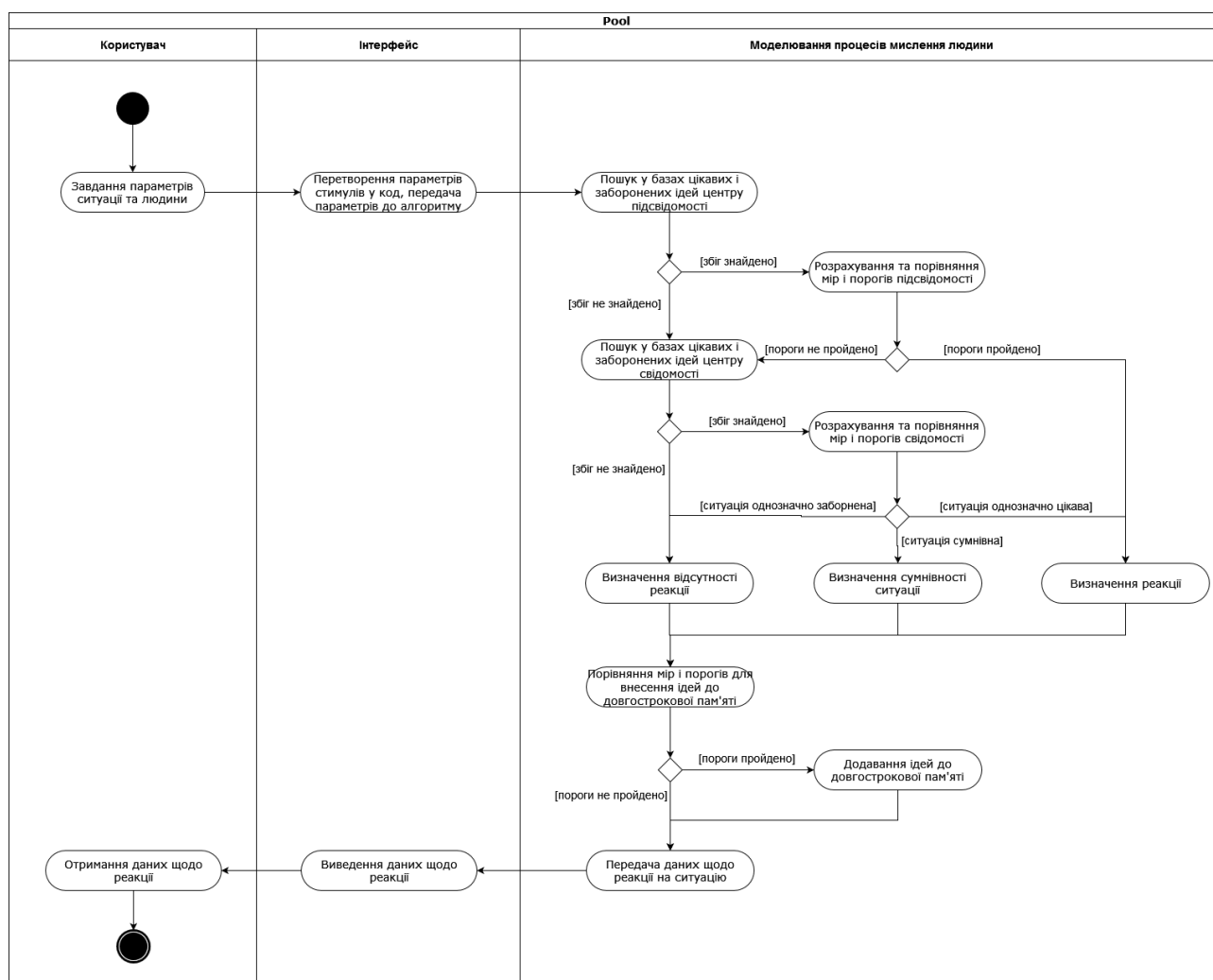


Рисунок 2.1 – Діаграма активностей

2.2.2 Діаграма прецедентів

Потік подій для розробленої діаграми прецедентів (рис. 2.2).

Крок 1. Користувач обирає суб'єкта експерименту. За необхідності користувач редагує індивідуальні характеристики людини (a , b , c , d , C , T_{i1} , T_{i2} , T_m , I_{min} , I_{max} , F_{max} , I_{mem} , F_{mem} , S_{mem} , I_+ , F_+ , S_+).

Крок 2. Користувач відтворює ситуацію та отримує результат роботи процесів мислення обраного суб'єкту.

Крок 3. Користувач за допомогою дошки із результататми попередніх експериментів із різними суб'єктами та/або ситуаціями спостерігає різницю між ними, що залежить від індивідуальних характеристик людини та її досвіду.

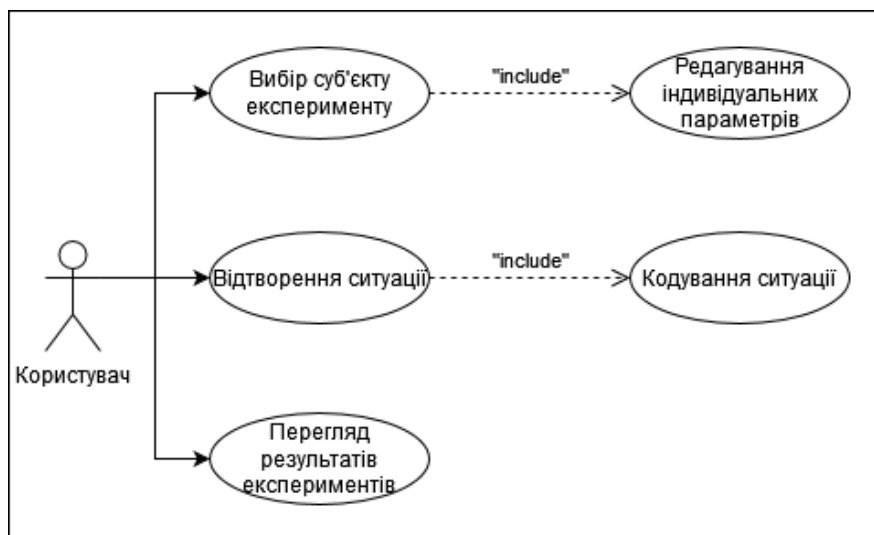


Рисунок 2.2 – Діаграма прецедентів

2.2.3 Діаграма послідовностей

Потік подій для розробленої діаграми послідовностей (рис. 2.3).

Крок 1. Користувач вводить дані щодо індивіду та ситуації до програмного інтерфейсу та підтверджує відтворення ситуації.

Крок 2. Програма отримує від інтерфейсу характеристики експерименту й опрацьовує поточні дані (ситуацію, індивідуальні характеристики людини).

Крок 3. Програма відправляє вихідні дані до інтерфейсу.

Крок 4. Користувач переглядає інформацію, виведену на дошку результатів експерименту на інтерфейсі.

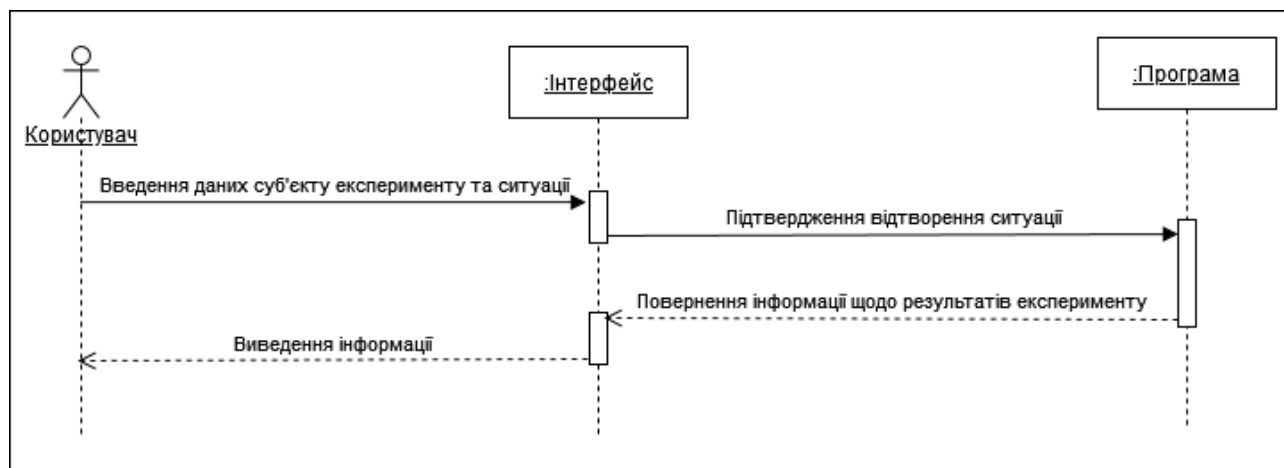


Рисунок 2.3 – Діаграма послідовностей

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

У даному розділі було розглянуто основні рішення із розробки класів, інтерфейсу користувача та алгоритмів відображення роботи процесів мислення людини.

3.1 Розробка алгоритму

О. Ю. Хренніков пропонує певну математичну модель процесів мислення людини [1], що використовує p -адичну метрику, побудовану К. Гензелем. Ця метрика представляє собою послідовність чисел, що відображає відстань між ідеями. Відстань вимірюється за формулою (3.1):

$$p_m(x, y) = \frac{1}{m^k} \quad (3.1)$$

де x, y – послідовності цифр з основою $m > 1$;

k – така, що x і y мають різні цифри на цій позиції.

Для визначення індивідуальних параметрів (констант і порогів) людини необхідне проведення певних тестів на психологічний стан людини, на результатах яких формуються припущення, перетворені у перелічені далі коефіцієнти [1].

Інформація J_0 , що надходить на аналізатори людини із зовнішнього середовища, представляється p -адичним числом [1]. У таблиці 3.1 наведено кодування ситуацій, прийняті у розроблюваному програмному забезпеченні.

J_0 передається як ідея-аттрактор J до множини процесорів мислення π керуючого центру підсвідомості (КЦП) та після обробки надсилається до аналізатору підсвідомості, де компаратор підсвідомості порівнює її з ідеями, що знаходяться у базах цікавих та заборонених ідей підсвідомості ($D_{\text{під},i}$ та $D_{\text{під},f}$ відповідно, де зберігаються дані щодо інстинктів: страху, голоду, агресії,

зняжовілості та інших (за класифікацією В. Мак-Дугалла [10]), і виявляє найближчі до неї ідеї [1].

Таблиця 3.1 – Кодування ситуації

Код	Стать	Колір волосся	Вік	Зріст	Предмет	Соціальний рівень
0	Жінка	Блондин(ка)	Менше 20	Низький	Яблуко	Бідний
1	Чоловік	Рудий(а)	20 - 30	Середній	Сумка	Середнього достатку
2		Шатен(ка)	30 - 45	Високий	Ніж	Багатий
3		Брюнет(ка)	45 - 60			
4			Більше 60			

Компаратор обчислює міру інтересу $I(J)$, міру протидії $F(J)$ та міру узгодженості $T(J)$ ідеї за формулами 3.2 - 3.4 [1]:

$$I(J) = \frac{1}{1 + p(J, D_i)}, \quad (3.2)$$

$$F(J) = \frac{1}{1 + p(J, D_f)}, \quad (3.3)$$

$$T(J) = a * I(J) - b * F(J), \quad (3.4)$$

де a, b – деякі параметри, що залежать від системи мислення.

Далі $T(J)$ порівнюється із порогом узгодженості T_{i1} і порогом фізичної реалізації T_{i2} . Якщо пороги було пройдено ($|T(J)| \geq T_{i1}$ і $|T(J)| \geq T_{i2}$), то J відправляється на реалізацію, в іншому випадку вона передається до керуючого центру свідомості (КЦС), а саме до оперативної пам'яті (ОП), де компаратор свідомості порівнює її з ідеями, що знаходяться у базах цікавих та заборонених ідей свідомості ($D_{св,i}$ та $D_{св,f}$ відповідно, де зберігаються ідеї, що протягом існування індивіду та отримання їм досвіду стали достатньо цікавими та/або забороненими, щоб бути збереженими до довгострокової пам'яті (ДП)) [1].

Компаратор обчислює $I(J)$, $F(J)$ і $T(J)$ за формулами 3.2 - 3.4, а також силу ідеї $S(J)$ за формулою 3.5 [1]:

$$S(J) = c * I(J) + d * F(J), \quad (3.5)$$

де c, d – деякі параметри, що залежать від системи мислення.

Далі $I(J)$ порівнюється із порогом мінімальної цікавості ідеї $I_{\min}$. Якщо поріг не було пройдено ($I < I_{\min}$), то J видаляється, інакше вона передається далі. Перед відправленням на реалізацію J має пройти ще ряд порогів [1]:

- якщо J одночасно не проходить пороги надзвичайної цікавості ідеї $I_{\max}$ і реалізації T_m ($I(J) < I_{\max}$ і $T(J) < T_m$), то вона видаляється;
- якщо J одночасно проходить T_m і не проходить поріг надзвичайної протидії $F_{\max}$ ($F(J) < F_{\max}$ і $T(J) \geq T_m$), то вона відправляється на реалізацію;
- якщо J одночасно проходить пороги $I_{\max}$ і $F_{\max}$ ($(I(J) \geq I_{\max}$ і $T(J) < T_m$) або ($F(J) \geq F_{\max}$ і $T(J) \geq T_m$)), то вона передається до області сумнівів O_d

Окрім цього, J може потрапити до ДП. Ідеї, що знаходяться у ДП, можуть втрачати свою важливість у часі за певною функцією $l_{\text{mem}}(t)$. Було прийнято використовувати закон, наведений у формулі 3.6 [1]:

$$l_{\text{mem}}(t) = e^{-Ct}, \quad (3.6)$$

де C – деяка константа, що залежить від системи мислення.

Якщо J проходить пороги для зберігання у ДП ($I(J) \geq I_{\text{mem}}$, або $F(J) \geq F_{\text{mem}}$, або $S(J) \geq S_{\text{mem}}$), то ідея додається до $D_{\text{св},i}$ або $D_{\text{св},f}$ відповідно до її характеристик. Якщо $I(J) \geq I_+$, або $F(J) \geq F_+$, або $S(J) \geq S_+$ (де $I_+ \geq I_{\text{mem}}$, і $F_+ \geq F_{\text{mem}}$, і $S_+ \geq S_{\text{mem}}$), то J додається у ДП і не змінює свою важливість у часі ($C = 0$). Обчислення $I(J)$, $F(J)$ і $T(J)$ у момент t відбувається за формулами 3.7 - 3.9 [1]:

$$I(t, J) = l_{\text{mem}}(t - t_0) * I(J), \quad (3.7)$$

$$F(t, J) = l_{\text{mem}}(t - t_0) * F(J), \quad (3.8)$$

$$S(t, J) = l_{\text{mem}}(t - t_0) * S(J), \quad (3.9)$$

де $I(J)$, $F(J)$, $S(J)$ – початкові значення міри інтересу, міру протидії та сили ідеї;
 t, t_0 – поточний момент часу і момент часу додавання ідеї до ДП.

Відповідно ідеї видаляються із ДП, якщо вони втратили свою важливість із часом ($I(t, J) \leq I_{\text{mem}}$, і $F(t, J) \leq F_{\text{mem}}$, і $S(t, J) \leq S_{\text{mem}}$) [1].

3.2 Основні рішення щодо розроблених класів

3.2.1 Перелічення

Для зручності розуміння та програмування алгоритму було створено 2 перелічення: Bases і Characteristics.

Bases (до якого входять: NO_BASE = -1, INSTINCT_INTERESTING, INSTINCT_FORBIDDEN, MEMORY_INTERESTING, MEMORY_FORBIDDEN) відображає, у якій саме базі знаходиться ідея (ніяка, $D_{\text{під},i}$, $D_{\text{під},f}$, $D_{\text{св},i}$, $D_{\text{св},f}$).

Characteristics (до якого входять: SEX = 0, HAIR_COLOR, AGE, CONSTITUTION, ITEM, SOCIAL_LEVEL) використовується для кодування пріоритетів індивіду (порядку позицій параметрів ситуації: статі, кольору волосся, віку, зрісту, предмету, соціального рівню).

3.2.2 Клас Reactions

Клас Reactions призначений для зберігання переліку реакцій людини на отримувані ідеї. Поля та методи класу наведено у таблиці 3.2.

Таблиця 3.2 – Клас Reactions

Поля та методи класу	Опис
1	2
private:	
map<unsigned int, string> reactions	Усі можливі реакції людини
public:	
Reactions()	Конструктор, що ініціалізує реакції
string getReaction(int id)	Повертає реакцію за ідентифікатором

3.2.3 Клас Knowledge

Клас Knowledge призначений для зберігання даних щодо ідеї (коду, реакції, мір інтересу та протидії ідеї, сили ідеї, момент часу додавання ідеї до бази ідей, константи С (що використовується у функції часу), змінної для визначення, яка характеристика використовується для порівняння для видалення ідеї). Поля та методи класу наведено у таблиці 3.3.

Таблиця 3.3 – Клас Knowledge

Поля та методи класу	Опис
1	2
private:	
string situation	Код ситуації
unsigned int reaction	Код реакції (із класу Reactions)
double I	Міра інтересу ідеї
double F	Міра протидії ідеї
double S	Сила ідеї
quint64 time	Момент часу додавання ідеї до бази ідей
double C	Константа для розрахування функції часу
bool comparedToS	Чи порівнюється із S, а не I або F
public:	
Knowledge(string _situation, unsigned int _reaction)	Конструктор для ідеї у базі підсвідомості
Knowledge(string _situation, unsigned int _reaction, double _I, double _F, double _S, quint64 _time, double _C, bool _comparedToS = false)	Конструктор для ідеї у базі свідомості
string getSituation()	Повертає код ситуації
unsigned int getReaction()	Повертає ідентифікатор реакції на ситуацію
bool getComparedToS()	Повертає, чи порівнюється із S, а не I або F
double l_mem()	Повертає результат функції часу
unsigned int needToBeDeletedAsTimePassed(Bases base, double threshold)	Повертає, чи втратила ідея свою важливість у часі

3.2.4 Клас KnowledgeBase

Клас KnowledgeBase призначений для зберігання переліку Knowledge, та представляє певну базу ідей ($D_{\text{під,і}}$, $D_{\text{під,ф}}$, $D_{\text{св,і}}$ або $D_{\text{св,ф}}$). Поля та методи класу наведено у таблиці 3.4.

Таблиця 3.4 – Клас KnowledgeBase

Поля та методи класу	Опис
1	2
private:	
vector<Knowledge> knowledge_base	Перелік ідей
public:	
KnowledgeBase()	Конструктор порожньої бази ідей
~KnowledgeBase()	Деструктор
vector<Knowledge> getKnowledgeBase()	Повертає перелік ідей
void addKnowledge(Knowledge knowledge)	Додає ідею до переліку ідей
Knowledge getKnowledge(unsigned int id)	Повертає ідею за позицією у переліку
void deleteKnowledge(unsigned int id)	Видаляє ідею за позицією у переліку

3.2.5 Клас Human

Клас Human призначений для зберігання баз ідей у КЦП і КЦС ($D_{\text{під},i}$, $D_{\text{під},f}$, $D_{\text{св},i}$ і $D_{\text{св},f}$), індивідуальних параметрів людини (a , b , c , d , C , T_{i1} , T_{i2} , T_m , I_{min} , I_{max} , F_{max} , I_{mem} , F_{mem} , S_{mem} , I_+ , F_+ , S_+) і пріоритетів (порядку позицій параметрів ситуації). Поля та методи класу наведено у таблиці 3.5.

Таблиця 3.5 – Клас Human

Поля та методи класу	Опис
1	2
private:	
string name	Ім'я людини та назва ситуації експерименту
KnowledgeBase *instinct_interest	База цікавих ідей підсвідомості
KnowledgeBase *instinct_forbidden	База заборонених ідей підсвідомості
KnowledgeBase *memory_interest	База цікавих ідей свідомості
KnowledgeBase *memory_forbidden	База заборонених ідей свідомості
double a, double b, double c, double d	Певні параметри мозку людини
double C	Певна константа мозку людини
double Ti_1	Поріг узгодженості ідеї (підсвідомість)
double Ti_2	Поріг фізичної реалізації (підсвідомість)
double Tm	Поріг реалізації (свідомість)
double I_min	Поріг мінімального інтересу ідеї
double I_max	Поріг надзвичайного інтересу ідеї
double F_max	Поріг надзвичайної протидії ідеї
double I_mem	Поріг інтересу для зберігання у ДП
double F_mem	Поріг протидії для зберігання у ДП
double S_mem	Поріг сили ідеї для зберігання у ДП
double I_plus	Поріг інтересу для зберігання у ДП без зміни у часі
double F_plus	Поріг протидії для зберігання у ДП без зміни у часі

Продовження табл. 3.5

1	2
double S_plus	Поріг сили ідеї для зберігання у ДП без зміни у часі
vector<unsigned int> priorities	Порядок важливості параметрів ситуації
public:	
Human(string _name, KnowledgeBase _iibase, KnowledgeBase _ifbase, KnowledgeBase _mibase, KnowledgeBase _mfbase, double _a, double _b, double _c, double _d, double _C, double _Ti_1, double _Ti_2, double _Tm, double _I_min, double _I_max, double _F_max, double _I_mem, double _F_mem, double _S_mem, double _I_plus, double _F_plus, double _S_plus, vector<unsigned int> _priorities)	Конструктор індивіду
~Human()	Деструктор
string getName(bool full = false)	Повертає name (за замовченням лише ім'я)
double geta()	Повертає a
double getb()	Повертає b
double getc()	Повертає c
double getd()	Повертає d
double getC()	Повертає C
double getTi1()	Повертає Ti_1
double getTi2()	Повертає Ti_2
double getTm()	Повертає Tm
double getImin()	Повертає I_min
double getImax()	Повертає I_max
double getFmax()	Повертає F_max
double getImem()	Повертає I_mem
double getFmem()	Повертає F_mem
double getSmem()	Повертає S_mem
double getIplus()	Повертає I_plus
double getFplus ()	Повертає F_plus
double getSplus ()	Повертає S_plus
void seta(double _a)	Встановлює a
void setb(double _b)	Встановлює b
void setc(double _c)	Встановлює c
void setd(double _d)	Встановлює d
void setTi1(double _Ti_1)	Встановлює Ti_1
void setTi2(double _Ti_2)	Встановлює Ti_2
void setTm(double _Tm)	Встановлює Tm
void setImin(double _I_min)	Встановлює I_min
void setImax(double _I_max)	Встановлює I_max
void setFmax(double _F_max)	Встановлює F_max
void setImem(double _I_mem)	Встановлює I_mem
void setFmem(double _F_mem)	Встановлює F_mem
void setSmem(double _S_mem)	Встановлює S_mem
void setIplus(double _I_plus)	Встановлює I_plus
void setFplus (double _F_plus)	Встановлює F_plus

Продовження табл. 3.5

1	2
<code>void setSplus (double _S_plus)</code>	Встановлює <code>S_plus</code>
<code>double compare(string code1, string code2)</code>	Повертає відстань між двома ситуаціями
<code>string checkMemory()</code>	Повертає результат перевірки, чи втратила якась ідея у ДП свою важливість і була видаленою
<code>string situationPriorities(string situation)</code>	Повертає кодування ситуації-параметру у порядку пріоритетів людини
<code>string prioritiesToString()</code>	Повертає <code>priorities</code> у вигляді рядка
<code>string result(string situation)</code>	Повертає результат роботи процесів мислення людини на ситуацію-параметр

3.2.6 Діаграма класів

Усі перелічені раніше класи було представлено графічно на рис. 3.1, де:

- `Bases` – інформаційне перелічення, що передає тип бази ідей;
- `Characteristics` – інформаційне перелічення, що використовується для кодування параметрів ситуацій;
- `Reactions` – база всіх можливих реакцій людини, що дозволяє витягнути інформацію щодо реакції за ідентифікатором;
- `Knowledge` – певна ідея, що належить людині;
- `KnowledgeBase` – база цікавих або заборонених ідей (перелік `Knowledge`) свідомості або підсвідомості людини;
- `Human` – індивід, до якого належать бази цікавих та заборонених ідей свідомості та підсвідомості (4 `KnowledgeBase`).

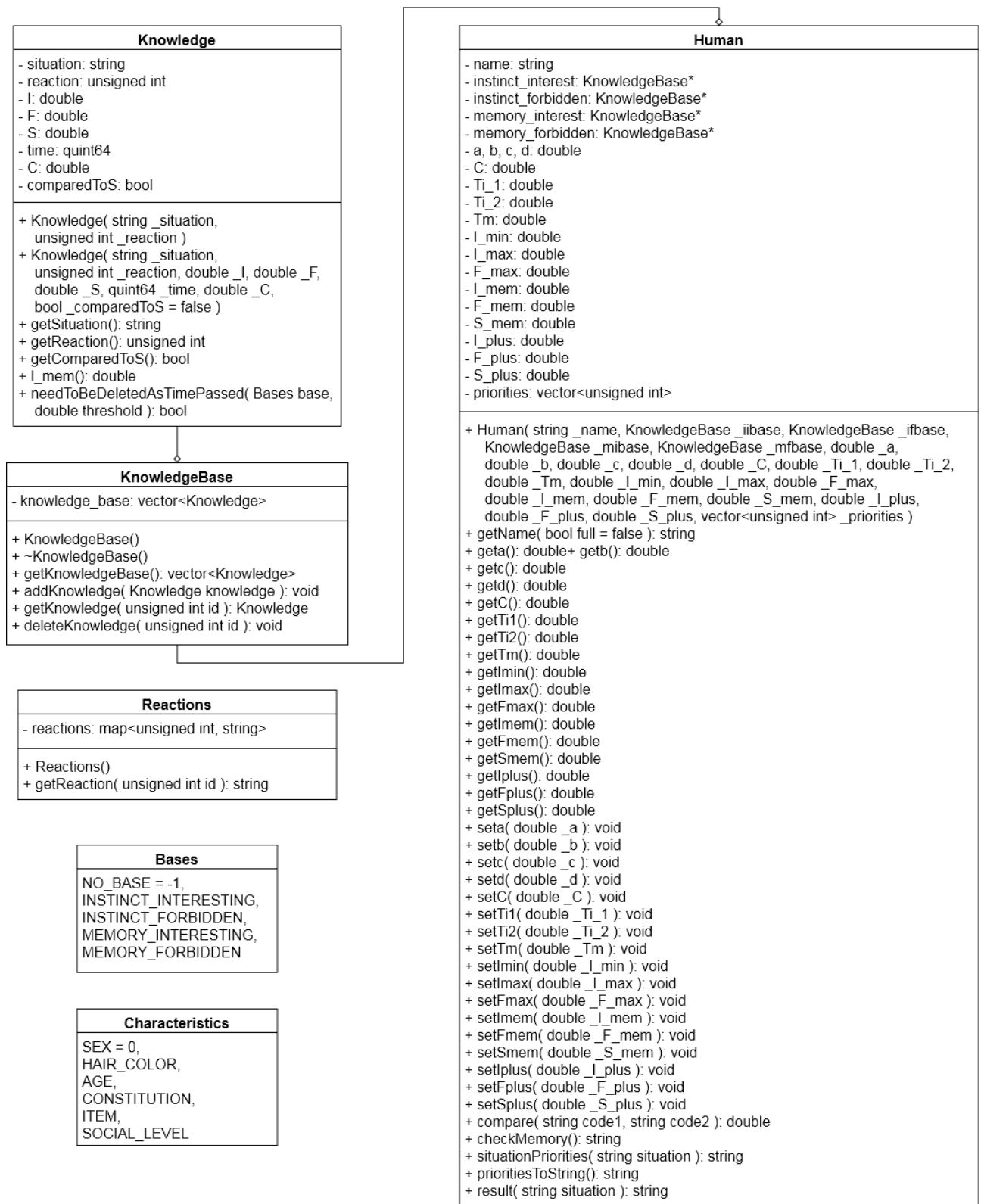


Рисунок 3.1 – Діаграма класів

3.3 Розробка графічного інтерфейсу

Розроблюваний інтерфейс повинен складатись із:

- блоку кодування ситуації;
- блоку вибору суб'єкту експерименту та редагування його індивідуальних параметрів;
- інформаційної дошки, до якої виводяться результати експериментів;
- кнопки відтворення ситуації.

При проектуванні форми програми було використано:

- випадаючі списки (для вибору індивіду та параметрів ситуації (стать, колір волосся, вік, зріст, предмет, соціальний рівень));
- поля для вводу чисел (встановлення індивідуальних параметрів людини: $a, b, c, d, C, T_{i1}, T_{i2}, T_m, I_{min}, I_{max}, F_{max}, I_{mem}, F_{mem}, S_{mem}, I_+, F_+, S_+$);
- текстове поле (у якості інформаційної дошки результатів);
- кнопку (для запуску алгоритму).

Для роботи графічного інтерфейсу було створено клас MainWindow. Поля та методи класу наведено у таблиці 3.6.

Таблиця 3.6 – Клас MainWindow

Поля та методи класу	Опис
1	2
private:	
MainWindow *ui	Об'єкт форми
vector<Human*> humans	Перелік індивідів
private slots:	
void On_listNames_currentIndexChanged(int index)	Викликається, якщо було змінено поточного суб'єкта експерименту
void On_btnAct_clicked()	Викликається при натисканні кнопки запуску алгоритму
public:	
MainWindow(QWidget *parent = nullptr)	Конструктор (ініціалізація форми та заповнення humans)
~MainWindow()	Деструктор
void setParameters()	Заповнює індивідуальні характеристики людини згідно до обраного суб'єкту експерименту

Остаточний вигляд форми наведено на рис. 3.2.

Моделювання розумової діяльності людини

Редактор ситуації:

Стать: Жінка Колір волосся: Блондин(ка)

Вік: Менше 20 Зріст: Високий

Предмет: Ніж Соціальний рівень: Бідний

Хід моделювання

Людина: Іван.
 Пріоритети розташовані: 430521.
 Отримано ідею 000220 (за пріоритетами 220000)!

Відбувається пошук у підсвідомості!
 $I = 0.9730$
 $F = 0.9730$
 $T = -0.4865$

Ситуація одночасно відноситься до бази цікавих (1xx220, за пріоритетами 2210xx) - можлива реакція 'злякався і викликав поліцію' - та бази заборонених (1xx220, за пріоритетами 2210xx) - можлива реакція 'злякався і втік' - ідей підсвідомості!
 Ситуація є більш забороненою, ніж цікавою!
 Ідея проходить поріг узгодженості!
 Ідея проходить поріг фізичної реалізації!

Реакція: Іван злякався і втік.
 Рішення було знайдено у бази заборонених ідей підсвідомості!

Людина
 Іван (ситуація страху)

Індивідуальні параметри людини

a	b	c	d	C, 10^{-6}
0,40	0,90	0,40	0,30	1,00

Ti_1	I_min	I_mem	I+
0,300	0,400	0,800	0,900

Ti_2	I_max	F_mem	F+
0,400	0,700	0,800	0,900

Tm	F_max	S_mem	S+
0,600	0,650	1,100	1,500

Код ситуації
 000220

Почати

Рисунок 3.2 – Форма моделювання розумової діяльності людини

4 КЕРІВНИЦТВО ПРОГРАМІСТА

У даному розділі було розглянуто призначення, умови застосування та характеристики програми, звертання до неї, початкові та вихідні дані.

4.1 Призначення та умови застосування програми

4.1.1 Призначення програми

Програмне забезпечення "Моделювання розумової діяльності людини" призначене для дослідження роботи процесів мислення людини за обраною моделлю.

4.1.2 Функції програми

Програмне забезпечення здійснює:

- завдання запрограмованих параметрів ситуації;
- вибір одного із запропонованих програмою індивидів із визначеними пріоритетами;
- завдання індивідуальних параметрів людини для поточного суб'єкту експерименту;
- алгоритмічний аналіз ситуації та виведення реакції на нього із проміжковими результатами;
- внесення змін до баз ідей свідомості.

4.1.3 Умови застосування програми

Для коректного функціонування застосунку необхідна версія ОС Windows 7 та вище.

4.2 Характеристики програми

Програмне виконана за допомогою мови програмування високого рівня C++ у середовищі розробки Qt Creator 5.0.1. Для проектування і розробки графічного інтерфейсу використовувались віджети бібліотеки Qt.

4.2.1 Структура програми

Програма складається із 2 користувацьких перелічень і 5 класів, 4 з яких – користувацькі, 1 – клас форм. Ієрархію класів, їх реалізації і файлів ресурсів проєкту наведено на рис. 4.1.

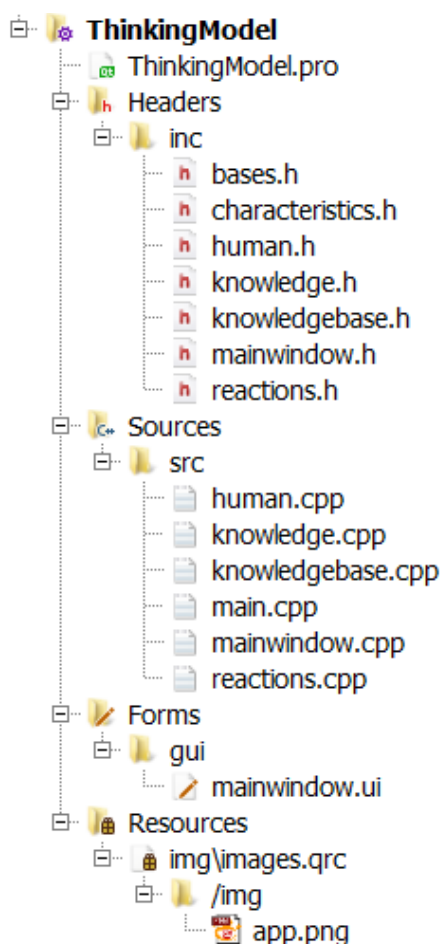


Рисунок 4.1 – Ієрархія проєкту

4.2.2 Алгоритм роботи програми

Під часу запуску програми ініціалізуються задані програмно дані щодо індивидів (бази ідей, пріоритети, індивідуальні характеристики), відображення та заповнення необхідних елементів для керування експериментом на розробленій формі. Під час завершення програми відбувається видалення всіх даних та закриття графічного інтерфейсу.

4.3 Звертання до програми

Для запуску програми необхідно попередньо упевнитися у відповідності характеристик комп'ютера, на який встановлена програма, до системних вимог програми. Після цього необхідно розархівувати папку із програмою та розташувати її у каталог, до якого має доступ оператор. Потрібно упевнитися у наявності всіх необхідних бібліотек для функціонування програми.

Звертання до програми передбачене одним способом: через виконуваний файл ThinkingModel.exe. Звертання до команди через командний рядок не передбачене, адже програма розроблена для роботи саме через графічний інтерфейс для завдання параметрів ситуації та людини.

4.4 Початкові та вихідні дані

Початкові дані представляють собою перелік індивидів, кожен із яких має бази цікавих та заборонених ідей свідомості і підсвідомості, пріоритети для кодування ситуацій та індивідуальні характеристики.

Вихідні дані представляють собою змінені бази цікавих і заборонених ідей свідомості, виведені на інформаційну дошку дані щодо роботи процесів мислення людини та її реакції на ситуацію.

5 КЕРІВНИЦТВО ОПЕРАТОРА

У даному розділі було розглянуто призначення програми, умови та процес її виконання.

5.1 Призначення програми

Програмне забезпечення "Моделювання розумової діяльності людини" призначене для дослідження роботи процесів мислення людини за обраною моделлю.

5.2 Умови виконання програми

5.2.1 Апаратні вимоги програми

Вимоги до комп'ютера, на якому виконується програмне забезпечення, наведено у п. 1.2.3.4 даної пояснювальної записки.

5.2.2 Вимоги до користувача

До користувачів висуваються наступні вимоги:

- знання української мови (мова інтерфейсу);
- базове розуміння алгоритму, що відтворюється;
- вільне орієнтування у значеннях індивідуальних параметрів людини

для експериментування та аналізу результатів.

5.3 Виконання програми

Програма запускається за допомогою подвійного натискання лівої клавіші миші на виконуваний файл програми або її ярлик ThinkingModel.exe, розархівований разом із іншими додатковими файлами у папці програмного

забезпечення. Потрібно упевнитися у наявності всіх необхідних бібліотек і виконання всіх системних вимог для повноцінного функціонування програми.

Після запуску виконуваного файлу з'являється головна форма програми, що свідчить про початок роботи із програмни забезпеченням. Головна форма програми зображена на рис. 5.1.

Рисунок 5.1 – Головна форма програми

Зліва розташовано редактор ситуації, у якому кодується майбутня ідея, що включає до себе параметри: стать, колір волосся, вік, зріст, предмет і соціальний рівень. Нижче розташовано інформаційну дошку, на яку буде виведено результати роботи процесів мислення людини.

Справа розташовано редактор суб'єкту експерименту, де обирається індивід (разом із певними запрограмованими пріоритетами) та редагуються його індивідуальні параметри.

У нижній частині форми розташовано кнопку "Почати", натискання якої запускає алгоритм обробки закодованої ситуації мозком обраної людини. Результати та проміжкові коментарі буде виведено на інформаційну дошку.

5.4 Приклад роботи програми

Для моделювання ситуації із баз ідей підсвідомості (ситуації страху) можна використати дані щодо індивідів Івана та Петра.

Іван та Петро мають у базі цікавих ідей підсвідомості ідею "1xx220" (де символ "х" позначає, що значення на цій позиції не є важливим), що розшифровується як "бідний високий чоловік із ножем", із реакцією на неї "викликати поліцію". У базі заборонених ідей підсвідомості така сама ідея "1xx220" збережена із реакцією "втекти". На прикладі двох суб'єктів експерименту можна побачити, як люди можуть по-різному реагувати на однакову ситуацію.

Відтворимо ситуацію "100220" для Івана та Петра відповідно. Індивіди мають однакові пріоритети до визначеної ситуації, тому для розрахунків відстаней між ідеями (формули 3.1 - 3.3) використовується кодування "221000". Ситуація повністю співпадає із ідеями у базах підсвідомості, тому міра інтересу I та міра протидії F дорівнюють 1.

Міра узгодженості T , розрахована за формулою 3.4, для Петра дорівнює 0.4, а для Івана – -0.5 (де додатне число значить, що переважає ідея із бази цікавих ідей, а від'ємне – із бази заборонених). Обидві ідеї проходять пороги узгодженості та фізичної реалізації підсвідомості (T_{i1} і T_{i2}) індивідів. Іван реагує на ситуацію через "втекти", а Петро – "викликати поліцію".

Результати відтворення ситуації наведено на рис. 5.2 - 5.3.

Моделювання розумової діяльності людини

Редактор ситуації:

Стать: Чоловік Колір волосся: Блондин(ка)

Вік: Менше 20 Зріст: Високий

Предмет: Ніж Соціальний рівень: Бідний

Хід моделювання

Людина: Іван.
Пріоритети розташовані: 430521.
Отримано ідею 100220 (за пріоритетами 221000)!

Відбувається пошук у підсвідомості!
I = 1.0000
F = 1.0000
T = -0.5000

Ситуація одночасно відноситься до бази цікавих (1xx220, за пріоритетами 2210xx) - можлива реакція 'злякався і викликав поліцію' - та бази заборонених (1xx220, за пріоритетами 2210xx) - можлива реакція 'злякався і втік' - ідей підсвідомості!
Ситуація є більш забороненою, ніж цікавою!
Ідея проходить поріг узгодженості!
Ідея проходить поріг фізичної реалізації!

Реакція: Іван злякався і втік.
Рішення було знайдено у базі заборонених ідей підсвідомості!

Людина
Іван (ситуація страху)

Індивідуальні параметри людини

a	b	c	d	C, 10 ⁻⁶
0,40	0,90	0,40	0,30	1,00

Ti_1	I_min	I_mem	I+
0,300	0,400	0,800	0,900

Ti_2	I_max	F_mem	F+
0,400	0,700	0,800	0,900

Tm	F_max	S_mem	S+
0,600	0,650	1,100	1,500

Код ситуації
100220

Почати

Рисунок 5.2 – Ситуація страху (Іван)

Моделювання розумової діяльності людини

Редактор ситуації:

Стать: Чоловік Колір волосся: Блондин(ка)

Вік: Менше 20 Зріст: Високий

Предмет: Ніж Соціальний рівень: Бідний

Хід моделювання

Людина: Петро.
Пріоритети розташовані: 430521.
Отримано ідею 100220 (за пріоритетами 221000)!

Відбувається пошук у підсвідомості!
I = 1.0000
F = 1.0000
T = 0.4000

Ситуація одночасно відноситься до бази цікавих (1xx220, за пріоритетами 2210xx) - можлива реакція 'злякався і викликав поліцію' - та бази заборонених (1xx220, за пріоритетами 2210xx) - можлива реакція 'злякався і втік' - ідей підсвідомості!
Ситуація є більш цікавою, ніж забороненою!
Ідея проходить поріг узгодженості!
Ідея проходить поріг фізичної реалізації!

Реакція: Петро злякався і викликав поліцію.
Рішення було знайдено у базі цікавих ідей підсвідомості!

Людина
Петро (ситуація страху)

Індивідуальні параметри людини

a	b	c	d	C, 10 ⁻⁶
0,90	0,50	0,40	0,30	1,00

Ti_1	I_min	I_mem	I+
0,300	0,400	0,800	0,900

Ti_2	I_max	F_mem	F+
0,400	0,700	0,800	0,900

Tm	F_max	S_mem	S+
0,600	0,650	1,100	1,500

Код ситуації
100220

Почати

Рисунок 5.3 – Ситуація страху (Петро)

Для моделювання ситуації із баз ідей свідомості можна використати дані щодо індивідів Сергія та Кирила.

Сергій має у базі цікавих ідей свідомості ідею "00xxx2", що розшифровується як "багата блондинка", із реакцією на неї "познайомитися". Ця ідея не втрачає свою важливість із часом. У базі заборонених ідей свідомості ідея "00xxx0", що розшифровується як "бідна блондинка", збережена із реакцією "ніяк не відреагувати". Ця ідея втрачає свою важливість із часом.

Відтворимо ситуацію "000000" для Сергія. Рішення не було знайдено у базах підсвідомості, тому здійснюється пошук у свідомості. Міра інтересу I дорівнює 0.973, міра протидії F – 1, міра узгодженості T – 0.0811. Ситуація проходить поріг мінімального інтересу $I_{\min}$ і є одночасно надзвичайно цікавою і надзвичайно забороненою (не проходить поріг реалізації T_m , але проходить поріг надзвичайного інтересу ідеї $I_{\max}$), тому вона відноситься до області сумнівів і реакція індивіду – "неможливість обрати рішення" (рис. 5.4).

Моделювання розумової діяльності людини

Редактор ситуації:

Стать: Жінка | Колір волосся: Блондин(ка)
Вік: Менше 20 | Зріст: Низький
Предмет: Яблуко | Соціальний рівень: Бідний

Хід моделювання

Ситуація не відноситься до жодної бази ідей у центрі підсвідомості!
Ідея не проходить поріг узгодженості!

Відбувається пошук у свідомості!

$I = 0.9730$
 $F = 1.0000$
 $T = 0.0811$
 $S = 0.6892$

Ситуація одночасно відноситься до бази цікавих (00xxx2, за пріоритетами 002xxx) - можлива реакція 'зацікавився цією людиною і підійшов познайомитися' - та бази заборонених (00xxx0, за пріоритетами 000xxx) - можлива реакція 'ніяк не відреагував на ситуацію' - ідей свідомості!
Ситуація є одночасно надзвичайно цікавою і надзвичайно забороненою, тому ситуація відноситься до області сумнівів!

Реакція: Сергій сумнівається і не може обрати рішення.

Людина
Сергій (ситуація сумнівів)

Індивідуальні параметри людини

a	b	c	d	C, 10^{-6}
0,70	0,60	0,40	0,30	1,00

Ti_1	I_min	I_mem	I+
0,300	0,400	0,900	0,980

Ti_2	I_max	F_mem	F+
0,400	0,700	0,900	0,980

Tm	F_max	S_mem	S+
0,600	0,650	1,600	1,800

Код ситуації
000000

Почати

Рисунок 5.4 – Ситуація сумнівів (Сергій)

Через певний проміжок часу ситуація "00xxx0" у базі заборонених ідей свідомості втратить свою важливість і буде видалена. Таким чином Сергій буде вже по-іншому реагувати на ту саму ситуацію.

Відтворимо ситуацію "000000" для Сергія. Рішення не було знайдено у базах підсвідомості, тому здійснюється пошук у свідомості. Міра інтересу I дорівнює 0.973, міра протидії F – 0, міра узгодженості T – 0.6811. Ситуація проходить поріг мінімального інтересу $I_{\min}$ і поріг реалізації T_m і не є надзвичайно забороненою (не проходить поріг надзвичайної протидії $F_{\max}$), тому реакція індивіду – "познайомитися" (рис. 5.5).

Модельовання розумової діяльності людини

Редактор ситуації:

Стать: Жінка | Колір волосся: Блондин(ка)

Вік: Менше 20 | Зріст: Низький

Предмет: Яблуко | Соціальний рівень: Бідний

Хід моделювання

Відбувається пошук у свідомості!
 Міра протидії 00xxx2 у базі заборонених ідей свідомості зменшилася із часом, тому її було видалено!
 $I = 0.9730$
 $F = 0.0000$
 $T = 0.6811$
 $S = 0.3892$

Ситуація відноситься до бази цікавих (00xxx2, за пріоритетами 002xxx) - можлива реакція 'зацікавився цією людиною і підійшов познайомитися' - ідей свідомості!
 Ідея проходить поріг реалізації і не є надзвичайно забороненою!
 Ситуація є більш цікавою, ніж забороненою!

Реакція: Сергій зацікавився цією людиною і підійшов познайомитися.

Ситуація є настільки цікавою, що вона додається до бази цікавих ідей свідомості!
 Рішення було знайдено у базі цікавих ідей свідомості!

Людина
 Сергій (ситуація сумнівів)

Індивідуальні параметри людини

a	b	c	d	C, 10^{-6}
0,70	0,60	0,40	0,30	1,00

Ti_1	I_min	I_mem	I+
0,300	0,400	0,900	0,980

Ti_2	I_max	F_mem	F+
0,400	0,700	0,900	0,980

Tm	F_max	S_mem	S+
0,600	0,650	1,600	1,800

Код ситуації
 000000

Почати

Рисунок 5.5 – Ситуація вирішених сумнівів (Сергій)

Кирило має у базі цікавих ідей свідомості ідею "03xxxx", що розшифровується як "брюнетка", із реакцією на неї "познайомитися". У базі заборонених ідей свідомості ідея "00xxxx", що розшифровується як "блондинка",

збережена із реакцією "ніяк не відреагувати". Обидві ідеї не втрачають свою важливість із часом.

Відтворимо ситуацію "000000" для Кирила. Рішення не було знайдено у базах підсвідомості, тому здійснюється пошук у свідомості. Міра інтересу I дорівнює 0.8571, міра протидії $F = 1$, міра узгодженості $T = 0.3857$. Ситуація проходить поріг мінімального інтересу $I_{\min}$, але не є надзвичайно цікавою (не проходить поріг надзвичайного інтересу ідеї $I_{\max}$), тому реакція індивіду – "ніяк не відреагувати" (рис. 5.6).

Модельовання розумової діяльності людини

Редактор ситуації:

Стать: Жінка | Колір волосся: Блондин(ка)

Вік: Менше 20 | Зріст: Низький

Предмет: Яблуко | Соціальний рівень: Бідний

Хід моделювання

Ідеї не проходять поріг узгодженості:

Відбувається пошук у свідомості!

$I = 0.8571$
 $F = 1.0000$
 $T = 0.3857$
 $S = 0.6429$

Ситуація одночасно відноситься до бази цікавих (03xxxx, за пріоритетами 03xxxx) - можлива реакція 'зацікавився цією людиною і підійшов познайомитися' - та бази заборонених (00xxxx, за пріоритетами 00xxxx) - можлива реакція 'ніяк не відреагував на ситуацію' - ідеї свідомості!

Ідея не проходить поріг реалізації і не є надзвичайно цікавою!

Реакція: Кирило ніяк не відреагував на ситуацію.

Ця ситуація вже є у базі заборонених ідей центру свідомості!

Рішення не було знайдено у жодній базі ідей центрів свідомості і підсвідомості!

Людина
 Кирило (ситуація протидії)

Індивідуальні параметри людини

a	b	c	d	C, 10^{-6}
0,80	0,30	0,40	0,30	1,00

Ti_1	I_min	I_mem	I+
0,300	0,400	0,950	0,980

Ti_2	I_max	F_mem	F+
0,400	0,900	0,950	0,980

Tm	F_max	S_mem	S+
0,500	0,900	1,600	1,800

Код ситуації
 000000

Почати

Рисунок 5.6 – Ситуація протидії (Кирило)

Таким чином, на ту ж саму ситуацію "000000" Сергій і Кирило відреагували по-різному: Сергій спочатку сумнівався, але через деякий час все ж прийняв рішення "познайомитися", тоді як Кирило не зацікавився достатньо і прийняв рішення "ніяк не відругувати".

Відтворимо ситуацію "030000" для Кирила. Рішення не було знайдено у базах підсвідомості, тому здійснюється пошук у свідомості. Міра інтересу I дорівнює 1, міра протидії $F = 0.8571$, міра узгодженості $T = 0.5429$. Ситуація проходить поріг мінімального інтересу $I_{\min}$ і поріг реалізації T_m і не є надзвичайно забороненою (не проходить поріг надзвичайної протидії ідеї $F_{\max}$), тому реакція індивіду – "познайомитися" (рис. 5.7).

Модельовання розумової діяльності людини

Редактор ситуації:

Стать: Жінка | Колір волосся: Брюнет(ка)

Вік: Менше 20 | Зріст: Низький

Предмет: Яблуко | Соціальний рівень: Бідний

Хід моделювання

Відбувається пошук у свідомості!

$I = 1.0000$
 $F = 0.8571$
 $T = 0.5429$
 $S = 0.6571$

Ситуація одночасно відноситься до бази цікавих (03xxxx, за пріоритетами 03xxxx) - можлива реакція 'зацікавився цією людиною і підійшов познайомитися' - та бази заборонених (00xxxx, за пріоритетами 00xxxx) - можлива реакція 'ніж не відреагував на ситуацію' - ідей свідомості! Ідея проходить поріг реалізації і не є надзвичайно забороненою! Ситуація є більш цікавою, ніж забороненою!

Реакція: Кирило зацікавився цією людиною і підійшов познайомитися.

Ця ситуація вже є у базі цікавих ідей свідомості!
 Рішення було знайдено у базі цікавих ідей свідомості!

Людина
 Кирило (ситуація протидії)

Індивідуальні параметри людини

a	b	c	d	C, 10^{-6}
0,80	0,30	0,40	0,30	1,00

Ti_1	I_min	I_mem	I+
0,300	0,400	0,950	0,980

Ti_2	I_max	F_mem	F+
0,400	0,900	0,950	0,980

Tm	F_max	S_mem	S+
0,500	0,900	1,600	1,800

Код ситуації
 030000

Почати

Рисунок 5.7 – Ситуація інтересу (Кирило)

На розглянутих прикладах можна наочно побачити ключові частини алгоритму процесів мислення людини та їх залежність від індивідуальних параметрів людини та її досвіду.

ВИСНОВКИ

Під час виконання дипломної кваліфікаційної роботи бакалавра було вирішено актуальну задачу створення програмного забезпечення для моделювання розумової діяльності людини на базі обраної математичної моделі мозку [1].

Для досягнення мети проєкту було проведено аналіз предметної області та виокремлено основні відмінності розроблюваної системи від існуючих аналогів, розроблено основні вимоги до проєкту та обрано межі програмної реалізації моделі.

На основі технічного завдання, наведеного у розділі 1, було спроектовано архітектуру та моделі програмного забезпечення (за допомогою UML-діаграм); обрано інструментарій розробки (мову програмування та середовище розробки); створено систему кодування вхідної інформації для відтворення; реалізовано необхідні програмні класи і користувацький інтерфейс, що задовольняють функціональним вимогам, а також алгоритм обробки ситуації за допомогою імітації процесів мислення людини.

Розроблена система відповідає вимогам щодо інформаційної повноти, програмної сумісності та надійності.

Практична значущість роботи полягає у можливості моделювання роботи мозку у відповідь на певну ситуацію для дослідження закономірностей процесів мислення та імітації розумової діяльності людини за допомогою зручного програмного забезпечення, що дозволяє експериментувати із різноманітними індивідуальними психофізіологічними параметрами людини.

Програмна документація включає до себе технічне завдання, керівництва оператора та програміста, текст розробленої програми.

Результати цієї роботи були представлені на науково-практичній конференції "Тиждень науки-2023" [11].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Khrennikov, A.Y. Modelling of psychological behavior on the basis of ultrametric mental space: Encoding of categories by balls. / A.Y. Khrennikov // P-Adic Num Ultramet Anal Appl. – 2010. – № 2. – P. 1–20: [Electronic resource]. – Access mode: <https://doi.org/10.1134/S2070046610010012>.
2. GPT-4 [Electronic resource]. – Access mode: <https://openai.com/product/gpt-4>.
3. OpenAI Help Center [Electronic resource] / Natalie. – Access mode: <https://help.openai.com/en/articles/6783457-what-is-chatgpt>.
4. Sophia Beta - Hanson Robotics [Electronic resource]. – Access mode: <https://www.hansonrobotics.com/sophia-2020>.
5. Prata, S. C++ Primer Plus, 6th Edition [Text] / S. Prata. – Addison-Wesley Professional, 2021. – 1200 p.
6. Lutz, M. Programming Python, 2nd Edition [Text] / M. Lutz. – O'Reilly Media, Inc., 2001. – 1255 p.
7. Energy Efficiency across Programming Languages: How Do Energy, Time, and Memory Relate? [Text] / [R. Pereira, M. Couto, F. Ribeiro et al.] // Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering : Conference on Systems, Programming, Languages, and Applications: Software for Humanity, Vancouver BC, 23-24 October 2017 : proceedings. – New York: Association for Computing Machinery, 2017. – P. 256-267.
8. Embedded Software Development Tools & Cross Platform IDE | Qt Creator [Electronic resource]. – Access mode: <https://www.qt.io/product/development-tools>.
9. Visual Studio 2022 IDE - Programming Tool for Software Developers [Electronic resource]. – Access mode: <https://visualstudio.microsoft.com/vs>.
10. McDougall, W. An introduction to social psychology [Text] / W. McDougall. – London: Methuen & Co. Ltd., 1919. – 324 p.
11. Карнаух, В. Д. Моделювання розумової діяльності людини [Текст] / В.

Д. Карнаух // Тези доповідей науково-практичної конференції, Запоріжжя, 24–28 квітня 2023 р. : Тиждень науки-2023, Запоріжжя, 24–28 квітня 2023 р. : тези доповідей. – Запоріжжя: НУ "Запорізька політехніка", 2023. У друці.

ДОДАТОК А
Текст програми

A.1 Код файла bases.h

```
#ifndef BASES_H
#define BASES_H

enum Bases
{
    NO_BASE = -1,
    INSTINCT_INTERESTING,
    INSTINCT_FORBIDDEN,
    MEMORY_INTERESTING,
    MEMORY_FORBIDDEN
};

#endif // BASES_H
```

A.2 Код файла characteristics.h

```
#ifndef CHARACTERISTICS_H
#define CHARACTERISTICS_H

enum Characteristics
{
    SEX = 0,
    HAIR_COLOR,
    AGE,
    CONSTITUTION,
    ITEM,
    SOCIAL_LEVEL
};

#endif // CHARACTERISTICS_H
```

A.3 Код файла reactions.h

```
#ifndef REACTIONS_H
#define REACTIONS_H

#include <map>
#include <string>
```

```

class Reactions
{
private:
    std::map<unsigned int, std::string> reactions;

public:
    Reactions();

    std::string getReaction( int id );
};

#endif // REACTIONS_H

```

A.4 Код файлу reactions.cpp

```

#ifndef REACTIONS_H
#define REACTIONS_H

#include <map>
#include <string>

class Reactions
{
private:
    std::map<unsigned int, std::string> reactions;

public:
    Reactions();

    std::string getReaction( int id );
};

#endif // REACTIONS_H #include "reactions.h"

//-----
Reactions::Reactions()
{
    reactions[0] = "ніяк не відреагував на ситуацію";
    reactions[1] = "зацікавився цією людиною і підійшов познайомитися";
    reactions[2] = "злякався і втік";
    reactions[3] = "злякався і викликав поліцію";
    reactions[4] = "сумнівається і не може обрати рішення";
}

```

```
//-----
std::string Reactions::getReaction( int id )
{
    return reactions[id];
}
```

A.5 Код файлу knowledge.h

```
#ifndef KNOWLEDGE_H
#define KNOWLEDGE_H

#include "bases.h"

#include <string>
#include <QDateTime>

class Knowledge
{
private:
    std::string situation;
    unsigned int reaction;

    //для довгострокової пам'яті

    double I;
    double F;
    double S;

    //для розрахування часу
    quint64 time;
    double C;
    bool comparedToS;

public:
    Knowledge( std::string _situation, unsigned int _reaction );
    Knowledge( std::string _situation, unsigned int _reaction, double _I, double
_F, double _S, quint64 _time, double _C, bool _comparedToS = false );

    std::string getSituation();
    unsigned int getReaction();
    bool getComparedToS();

    double l_mem();
```

```
    bool needToBeDeletedAsTimePassed( Bases base, double threshold );
};
```

```
#endif // KNOWLEDGE_H
```

A.6 Код файла knowledge.cpp

```
#include "knowledge.h"
```

```
//-----
Knowledge::Knowledge( std::string _situation, unsigned int _reaction )
{
    situation = _situation;
    reaction = _reaction;
```

```
    I = 0;
    F = 0;
    S = 0;
    time = 0;
    C = 0;
    comparedToS = false;
}
```

```
//-----
Knowledge::Knowledge( std::string _situation, unsigned int _reaction, double
_I, double _F, double _S, quint64 _time, double _C, bool _comparedToS )
{
    situation = _situation;
    reaction = _reaction;

    I = _I;
    F = _F;
    S = _S;
    time = _time;
    C = _C;
    comparedToS = _comparedToS;
}
```

```
//-----
std::string Knowledge::getSituation()
{
    return situation;
}
```

```

//-----
unsigned int Knowledge::getReaction()
{
    return reaction;
}

//-----
bool Knowledge::getComparedToS()
{
    return comparedToS;
}

//-----
double Knowledge::l_mem()
{
    return      std::exp(      -C      *      (
QDateTime::currentDateTime().currentMSecsSinceEpoch() - time ) );
}

//-----
bool Knowledge::needToBeDeletedAsTimePassed( Bases base, double
threshold )
{
    if( comparedToS )
    {
        if( l_mem() * S < threshold )
            return true;
        else
            return false;
    }
    else
    {
        switch( base )
        {
            case MEMORY_INTERESTING:
            {
                if( l_mem() * I < threshold )
                    return true;
                else
                    return false;
            }
            case MEMORY_FORBIDDEN:
            {
                if( l_mem() * F < threshold )
                    return true;

```

```

        else
            return false;
    }
    default:
    {
        return false;
    }
}
}
}

```

A.7 Код файла knowledgebase.h

```

#ifndef KNOWLEDGEBASE_H
#define KNOWLEDGEBASE_H

#include "knowledge.h"

#include <vector>

class KnowledgeBase
{
private:
    std::vector<Knowledge> knowledge_base;

public:
    KnowledgeBase();
    ~KnowledgeBase();

    std::vector<Knowledge> getKnowledgeBase();

    void addKnowledge( Knowledge knowledge );
    Knowledge getKnowledge( unsigned int id );
    void deleteKnowledge( unsigned int id );
};

#endif // KNOWLEDGEBASE_H

```

A.8 Код файла knowledgebase.cpp

```

#include "knowledgebase.h"

//-----

```

```

KnowledgeBase::KnowledgeBase()
{

}

//-----
std::vector<Knowledge> KnowledgeBase::getKnowledgeBase( void )
{
    return knowledge_base;
}

//-----
void KnowledgeBase::addKnowledge( Knowledge knowledge )
{
    knowledge_base.push_back( knowledge );
}

//-----
Knowledge KnowledgeBase::getKnowledge( unsigned int id )
{
    return knowledge_base[id];
}

//-----
void KnowledgeBase::deleteKnowledge( unsigned int id )
{
    knowledge_base.erase( std::next( knowledge_base.begin(), id ) );
}

//-----
KnowledgeBase::~~KnowledgeBase()
{
    knowledge_base.clear();
}

```

A.9 Код файла human.h

```

#ifndef HUMAN_H
#define HUMAN_H

#include "knowledgebase.h"

#include <vector>

```

```

class Human
{
private:
    std::string name;

    //бази даних цікавих та заборонених ідей центрів свідомості та
    підсвідомості
    KnowledgeBase *instinct_interest;
    KnowledgeBase *instinct_forbidden;

    KnowledgeBase *memory_interest;
    KnowledgeBase *memory_forbidden;

    //певні параметри мозку
    double a;
    double b;
    double c;
    double d;
    double C;

    //пороги
    double Ti_1; //узгодженості ідеї (підсвідомість)
    double Ti_2; //фізичної реалізації ідеї (підсвідомість)
    double Tm; //реалізації ідеї (свідомість)

    double I_min; //мінімального інтересу ідеї
    double I_max; //надзвичайного інтересу ідеї
    double F_max; //надзвичайної протидії ідеї

    double I_mem; //інтересу для зберігання у довгостроковій пам'яті
    double F_mem; //протидії для зберігання у довгостроковій пам'яті
    double S_mem; //сили ідеї для зберігання у довгостроковій пам'яті

    double I_plus; //інтересу для зберігання у довгостроковій пам'яті без
    зміни у часі
    double F_plus; //протидії для зберігання у довгостроковій пам'яті без
    зміни у часі
    double S_plus; //сили ідеї для зберігання у довгостроковій пам'яті без
    зміни у часі

    //пріоритети параметрів ситуації
    std::vector<unsigned int> priorities;

public:

```

```

        Human( std::string _name, KnowledgeBase _iibase, KnowledgeBase _ifbase,
KnowledgeBase _mibase, KnowledgeBase _mfbase,
                double _a, double _b, double _c, double _d, double _C, double _Ti_1,
double _Ti_2, double _Tm,
                double _I_min, double _I_max, double _F_max, double _I_mem, double
_F_mem, double _S_mem,
                double _I_plus, double _F_plus, double _S_plus, std::vector<unsigned
int> _priorities );
        ~Human();

//getters
std::string getName( bool full = false );
double geta();
double getb();
double getc();
double getd();
double getC();
double getTi1();
double getTi2();
double getTm();
double getImin();
double getImax();
double getFmax();
double getImem();
double getFmem();
double getSmem();
double getIplus();
double getFplus();
double getSplus();

//setters
void seta( double _a );
void setb( double _b );
void setc( double _c );
void setd( double _d );
void setC( double _C );
void setTi1( double _Ti_1 );
void setTi2( double _Ti_2 );
void setTm( double _Tm );
void setImin( double _I_min );
void setImax( double _I_max );
void setFmax( double _F_max );
void setImem( double _I_mem );
void setFmem( double _F_mem );
void setSmem( double _S_mem );

```

```

void setIplus( double _I_plus );
void setFplus( double _F_plus );
void setSplus( double _S_plus );

double compare( std::string code1, std::string code2 );
std::string checkMemory();

std::string situationPriorities( std::string situation );
std::string prioritiesToString();

std::string result( std::string situation );
};

#endif // HUMAN_H

```

A.10 Код файла human.cpp

```

#include "human.h"
#include "reactions.h"

#include <sstream>
#include <iomanip>

//-----
Human::Human( std::string _name, KnowledgeBase _iibase, KnowledgeBase
_ifbase, KnowledgeBase _mibase, KnowledgeBase _mfbase,
             double _a, double _b, double _c, double _d, double _C, double _Ti_1,
double _Ti_2, double _Tm,
             double _I_min, double _I_max, double _F_max, double _I_mem,
double _F_mem, double _S_mem,
             double _I_plus, double _F_plus, double _S_plus, std::vector<unsigned
int> _priorities )
{
    name = _name;

    instinct_interest = new KnowledgeBase( _iibase );
    instinct_forbidden = new KnowledgeBase( _ifbase );
    memory_interest = new KnowledgeBase( _mibase );
    memory_forbidden = new KnowledgeBase( _mfbase );

    a = _a;
    b = _b;
    c = _c;
    d = _d;

```

```

C = _C;

Ti_1 = _Ti_1;
Ti_2 = _Ti_2;
Tm = _Tm;

I_min = _I_min;
I_max = _I_max;
F_max = _F_max;

I_mem = _I_mem;
F_mem = _F_mem;
S_mem = _S_mem;

I_plus = _I_plus;
F_plus = _F_plus;
S_plus = _S_plus;

priorities = _priorities;
}

//-----
std::string Human::getName( bool full )
{
    if( full )
        return name;
    else
        return name.substr( 0, name.find( ' ' ) );
}

//-----
double Human::geta()
{
    return a;
}

//-----
double Human::getb()
{
    return b;
}

//-----
double Human::getc()
{

```

```
    return c;
}

//-----
double Human::getd()
{
    return d;
}

//-----
double Human::getC()
{
    return C;
}

//-----
double Human::getTi1()
{
    return Ti_1;
}

//-----
double Human::getTi2()
{
    return Ti_2;
}

//-----
double Human::getTm()
{
    return Tm;
}

//-----
double Human::getImin()
{
    return I_min;
}

//-----
double Human::getImax()
{
    return I_max;
}
```

```
//-----  
double Human::getFmax()  
{  
    return F_max;  
}  
  
//-----  
double Human::getImem()  
{  
    return I_mem;  
}  
  
//-----  
double Human::getFmem()  
{  
    return F_mem;  
}  
  
//-----  
double Human::getSmem()  
{  
    return S_mem;  
}  
  
//-----  
double Human::getIplus()  
{  
    return I_plus;  
}  
  
//-----  
double Human::getFplus()  
{  
    return F_plus;  
}  
  
//-----  
double Human::getSplus()  
{  
    return S_plus;  
}  
  
//-----  
void Human::seta( double _a )  
{
```

```
    a = _a;
}

//-----
void Human::setb( double _b )
{
    b = _b;
}

//-----
void Human::setc( double _c )
{
    c = _c;
}

//-----
void Human::setd( double _d )
{
    d = _d;
}

//-----
void Human::setC( double _C )
{
    C = _C;
}

//-----
void Human::setTi1( double _Ti_1 )
{
    Ti_1 = _Ti_1;
}

//-----
void Human::setTi2( double _Ti_2 )
{
    Ti_2 = _Ti_2;
}

//-----
void Human::setTm( double _Tm )
{
    Tm = _Tm;
}
```

```
//-----  
void Human::setImin( double _I_min )  
{  
    I_min = _I_min;  
}  
  
//-----  
void Human::setImax( double _I_max )  
{  
    I_max = _I_max;  
}  
  
//-----  
void Human::setFmax( double _F_max )  
{  
    F_max = _F_max;  
}  
  
//-----  
void Human::setImem( double _I_mem )  
{  
    I_mem = _I_mem;  
}  
  
//-----  
void Human::setFmem( double _F_mem )  
{  
    F_mem = _F_mem;  
}  
  
//-----  
void Human::setSmem( double _S_mem )  
{  
    S_mem = _S_mem;  
}  
  
//-----  
void Human::setIplus( double _I_plus )  
{  
    I_plus = _I_plus;  
}  
  
//-----  
void Human::setFplus( double _F_plus )  
{
```

```

    F_plus = _F_plus;
}

//-----
void Human::setSplus( double _S_plus )
{
    S_plus = _S_plus;
}

//-----
double Human::compare( std::string code1, std::string code2 )
{
    if( code1 == code2 )
        return 0;

    for( unsigned int i = 0; i < (unsigned int)( priorities.size() ); i++ )
    {
        //неважливий параметр
        if( code1[priorities[i]] != 'x' )
        {
            if( code1[priorities[i]] != code2[priorities[i]] )
            {
                return ( pow( code1.size(), -(int)i ) );
            }
        }
    }
    return 0;
}

//-----
std::string Human::checkMemory()
{
    unsigned int index = 0;
    std::string result = "";
    while( index < memory_interest->getKnowledgeBase().size() )
    {
        if( memory_interest->getKnowledgeBase()[index].getComparedToS() )
        {
            if(
                memory_interest-
>getKnowledgeBase()[index].needToBeDeletedAsTimePassed(
MEMORY_INTERESTING, S_mem ) )
            {
                result += "Сила ідеї " + memory_interest-
>getKnowledgeBase()[index].getSituation() +

```

" у базі цікавих ідей свідомості зменшилася із часом, тому її було видалено!\n";

```

        memory_interest->deleteKnowledge( index );
    }
    else
    {
        index++;
    }
}
else
{
    if(
memory_interest-
>getKnowledgeBase()[index].needToBeDeletedAsTimePassed(
MEMORY_INTERESTING, I_mem ) )
    {
        result += "Міра інтересу ідеї " + memory_interest-
>getKnowledgeBase()[index].getSituation() +
        " у базі цікавих ідей свідомості зменшилася із часом, тому її
було видалено!\n";

```

```

        memory_interest->deleteKnowledge( index );
    }
    else
    {
        index++;
    }
}
}

index = 0;
while( index < memory_forbidden->getKnowledgeBase().size() )
{
    if( memory_forbidden->getKnowledgeBase()[index].getComparedToS() )
    {
        if(
memory_forbidden-
>getKnowledgeBase()[index].needToBeDeletedAsTimePassed(
MEMORY_FORBIDDEN, S_mem ) )
        {
            result += "Сила ідеї " + memory_interest-
>getKnowledgeBase()[index].getSituation() +
            " у базі заборонених ідей свідомості зменшилася із часом,
тому її було видалено!\n";

            memory_forbidden->deleteKnowledge( index );

```

```

        }
        else
        {
            index++;
        }
    }
    else
    {
        if(
memory_forbidden-
>getKnowledgeBase()[index].needToBeDeletedAsTimePassed(
MEMORY_FORBIDDEN, F_mem ) )
        {
            result += "Міра протидії " + memory_interest-
>getKnowledgeBase()[index].getSituation() +
            " у базі заборонених ідей свідомості зменшилася із часом,
тому її було видалено!\n";

            memory_forbidden->deleteKnowledge( index );
        }
        else
        {
            index++;
        }
    }
}

return result;
}

//-----
std::string Human::situationPriorities( std::string situation )
{
    std::string situationPriority;
    for( unsigned int i = 0; i < priorities.size(); i++ )
    {
        situationPriority += situation[priorities[i]];
    }

    return situationPriority;
}

//-----
std::string Human::prioritiesToString()
{
    std::string stringPriorities;

```

```

for( unsigned int i = 0; i < priorities.size(); i++ )
{
    stringPriorities += std::to_string( priorities[i] );
}

return stringPriorities;
}

//-----
std::string Human::result( std::string situation )
{
    std::ostringstream formatStringDouble;
    formatStringDouble << std::fixed;
    formatStringDouble << std::setprecision(4);

    std::string result = "-----\n";
    result += "Отримано ідею " + situation + " (за пріоритетами " +
situationPriorities( situation ) + ")!\n\n";
    Reactions *reactions = new Reactions();

    //пошук у підсвідомості

    result += "Відбувається пошук у підсвідомості!\n";

    //для інтересу
    int index_interest = -1;
    double max_match_interest = 1;

    for( unsigned int i = 0; i < (unsigned int)( instinct_interest-
>getKnowledgeBase().size() ); i++ )
    {
        double p = compare( instinct_interest-
>getKnowledgeBase()[i].getSituation(), situation );
        if( p < max_match_interest )
        {
            index_interest = i;
            max_match_interest = p;
        }
    }

    //для протидії
    int index_forbidden = -1;
    double max_match_forbidden = 1;

```

```

        for( unsigned int i = 0; i < (unsigned int)( instinct_forbidden-
>getKnowledgeBase().size() ); i++ )
        {
            double p = compare( instinct_forbidden-
>getKnowledgeBase()[i].getSituation(), situation );
            if( p < max_match_forbidden )
            {
                index_forbidden = i;
                max_match_forbidden = p;
            }
        }

```

//порівняння за мірою узгодженості T

```
int index_reaction = -1;
```

```
Bases index_base = NO_BASE;
```

```
double I = ( index_interest != -1 ) ? ( 1 / ( 1 + max_match_interest ) ) : 0;
```

```
double F = ( index_forbidden != -1 ) ? ( 1 / ( 1 + max_match_forbidden ) ) : 0;
```

```
double T = a * I - b * F;
```

```
formatStringDouble << I;
```

```
result += "I = " + formatStringDouble.str() + "\n";
```

```
formatStringDouble.str("");
```

```
formatStringDouble << F;
```

```
result += "F = " + formatStringDouble.str() + "\n";
```

```
formatStringDouble.str("");
```

```
formatStringDouble << T;
```

```
result += "T = " + formatStringDouble.str() + "\n\n";
```

```
formatStringDouble.str("");
```

```

if( index_interest == -1 && index_forbidden == -1 )
{
    result += "Ситуація не відноситься до жодної бази ідей у центрі
підсвідомості!\n";
    index_reaction = 0;
}
else if( index_interest != -1 && index_forbidden != -1 )
{
    result += "Ситуація одночасно відноситься до бази цікавих (" +
instinct_interest->getKnowledgeBase()[index_interest].getSituation() +

```

```

        ", за пріоритетами " + situationPriorities( instinct_interest-
>getKnowledgeBase()[index_interest].getSituation() ) +
        ") - можлива реакція '" + reactions->getReaction( instinct_interest-
>getKnowledgeBase()[index_interest].getReaction() ) +
        "' - та бази заборонених (" + instinct_forbidden-
>getKnowledgeBase()[index_forbidden].getSituation() +
        ", за пріоритетами " + situationPriorities( instinct_forbidden-
>getKnowledgeBase()[index_forbidden].getSituation() ) +
        ") - можлива реакція '" + reactions->getReaction(
instinct_forbidden->getKnowledgeBase()[index_forbidden].getReaction() ) + "' - ідей
підсвідомості!\n";

```

```

    if( T > 0 )
    {
        index_base = INSTINCT_INTERESTING;
        index_reaction = instinct_interest-
>getKnowledgeBase()[index_interest].getReaction();
        result += "Ситуація є більш цікавою, ніж забороненою!\n";
    }
    else if ( T < 0 )
    {
        index_base = INSTINCT_FORBIDDEN;
        index_reaction = instinct_forbidden-
>getKnowledgeBase()[index_forbidden].getReaction();
        result += "Ситуація є більш забороненою, ніж цікавою!\n";
    }
    else
    {
        result += "Ситуація є неоднозначною!\n";
        index_reaction = 0;
    }
}
else if( index_interest != -1 )
{
    result += "Ситуація відноситься до бази цікавих (" + instinct_interest-
>getKnowledgeBase()[index_interest].getSituation() +
        ", за пріоритетами " + situationPriorities( instinct_interest-
>getKnowledgeBase()[index_interest].getSituation() ) +
        ") - можлива реакція '" + reactions->getReaction( instinct_interest-
>getKnowledgeBase()[index_interest].getReaction() ) +
        "' - ідей підсвідомості!\n";

    index_base = INSTINCT_INTERESTING;
    index_reaction = instinct_interest-
>getKnowledgeBase()[index_interest].getReaction();

```

```

        result += "Ситуація є цікавою!\n";
    }
    else
    {
        result += "Ситуація відноситься до бази заборонених (" +
instinct_forbidden->getKnowledgeBase()[index_forbidden].getSituation() +
        ", за пріоритетами " + situationPriorities( instinct_forbidden-
>getKnowledgeBase()[index_forbidden].getSituation() ) +
        ") - можлива реакція " + reactions->getReaction(
instinct_forbidden->getKnowledgeBase()[index_forbidden].getReaction() ) +
        "' - ідей підсвідомості!\n";

        index_base = INSTINCT_FORBIDDEN;
        index_reaction = instinct_forbidden-
>getKnowledgeBase()[index_forbidden].getReaction();
        result += "Ситуація є забороненою!\n";
    }

    //пороги підсвідомості
    bool instinct_reaction = false;

    if( abs( T ) >= Ti_1 )
    {
        result += "Ідея проходить поріг узгодженості!\n";

        if( abs( T ) >= Ti_2 )
        {
            result += "Ідея проходить поріг фізичної реалізації!\n\n";
            result += "Реакція: " + getName() + " " + reactions->getReaction(
index_reaction ) + ".\n";

            instinct_reaction = true;
        }
        else
        {
            result += "Ідея не проходить поріг фізичної реалізації!\n";
        }
    }
    else
    {
        result += "Ідея не проходить поріг узгодженості!\n";
    }

    //пошук у свідомості

```

```

if( !instinct_reaction )
{
    result += "\n-----\n";
    result += "Відбувається пошук у свідомості!\n";
    result += checkMemory() + "\n";

    //для інтересу
    index_interest = -1;
    max_match_interest = 1;

    for( unsigned int i = 0; i < (unsigned int)( memory_interest-
>getKnowledgeBase().size() ); i++ )
    {
        double p = compare( memory_interest-
>getKnowledgeBase()[i].getSituation(), situation );
        if( p < max_match_interest )
        {
            index_interest = i;
            max_match_interest = p;
        }
    }

    //для протидії
    index_forbidden = -1;
    max_match_forbidden = 1;

    for( unsigned int i = 0; i < (unsigned int)( memory_forbidden-
>getKnowledgeBase().size() ); i++ )
    {
        double p = compare( memory_forbidden-
>getKnowledgeBase()[i].getSituation(), situation );
        if( p < max_match_forbidden )
        {
            index_forbidden = i;
            max_match_forbidden = p;
        }
    }

    //порівняння за мірою узгодженості Т та силою ідеї S
    index_reaction = -1;
    index_base = NO_BASE;

    bool doubtful = false;
    bool min_interest = true;

```

```

I = ( index_interest != -1 ) ? ( 1 / ( 1 + max_match_interest ) ) : 0;
F = ( index_forbidden != -1 ) ? ( 1 / ( 1 + max_match_forbidden ) ) : 0;
T = a * I - b * F;
double S = c * I + d * F;

```

```

formatStringDouble << I;
result += "I = " + formatStringDouble.str() + "\n";
formatStringDouble.str("");

```

```

formatStringDouble << F;
result += "F = " + formatStringDouble.str() + "\n";
formatStringDouble.str("");

```

```

formatStringDouble << T;
result += "T = " + formatStringDouble.str() + "\n";
formatStringDouble.str("");

```

```

formatStringDouble << S;
result += "S = " + formatStringDouble.str() + "\n\n";
formatStringDouble.str("");

```

```

if( index_interest == -1 && index_forbidden == -1 )
{
    result += "Ситуація не відноситься до жодної бази ідей у центрі
свідомості!\n";
    index_reaction = 0;
}
else if( index_interest != -1 && index_forbidden != -1 )
{
    result += "Ситуація одночасно відноситься до бази цікавих (" +
memory_interest->getKnowledgeBase()[index_interest].getSituation() +
        ", за пріоритетами " + situationPriorities( memory_interest-
>getKnowledgeBase()[index_interest].getSituation() ) +
        ") - можлива реакція " + reactions->getReaction(
memory_interest->getKnowledgeBase()[index_interest].getReaction() ) +
        " - та бази заборонених (" + memory_forbidden-
>getKnowledgeBase()[index_forbidden].getSituation() +
        ", за пріоритетами " + situationPriorities( memory_forbidden-
>getKnowledgeBase()[index_forbidden].getSituation() ) +
        ") - можлива реакція " + reactions->getReaction(
memory_forbidden->getKnowledgeBase()[index_forbidden].getReaction() ) + " - ідей
свідомості!\n";
}

```

```

if( I < I_min )
{
    result += "Ідея не проходить поріг мінімальної інтересу ідеї!\n";
    index_reaction = 0;
    min_interest = false;
}
else
{
    if( ( T < Tm ) && ( I < I_max ) )
    {
        result += "Ідея не проходить поріг реалізації і не є надзвичайно
цікавою!\n";
        index_reaction = 0;
    }
    else if( ( T >= Tm ) && ( F < F_max ) )
    {
        result += "Ідея проходить поріг реалізації і не є надзвичайно
забороненою!\n";

        if( T > 0 )
        {
            index_base = MEMORY_INTERESTING;
            index_reaction = memory_interest-
>getKnowledgeBase()[index_interest].getReaction();
            result += "Ситуація є більш цікавою, ніж забороненою!\n";
        }
        else if ( T < 0 )
        {
            index_base = MEMORY_FORBIDDEN;
            index_reaction = memory_forbidden-
>getKnowledgeBase()[index_forbidden].getReaction();
            result += "Ситуація є більш забороненою, ніж цікавою!\n";
        }
        else
        {
            result += "Ситуація є неоднозначною!\n";
            index_reaction = 0;
        }
    }
    else //if( ( F >= F_max ) && ( I >= I_max ) )
    {
        result += "Ситуація є одночасно надзвичайно цікавою і
надзвичайно забороненою, тому ситуація відноситься до області сумнівів!\n";
        doubtful = true;
        index_reaction = 4;
    }
}

```

```

    }
  }
}
else if( index_interest != -1 )
{
    result += "Ситуація відноситься до бази цікавих (" +
memory_interest->getKnowledgeBase()[index_interest].getSituation() +
        ", за пріоритетами " + situationPriorities( memory_interest-
>getKnowledgeBase()[index_interest].getSituation() ) +
        ") - можлива реакція " + reactions->getReaction(
memory_interest->getKnowledgeBase()[index_interest].getReaction() ) +
        "' - ідей свідомості!\n";

    if( I < I_min )
    {
        result += "Ідея не проходить поріг мінімальної інтересу ідеї!\n";
        index_reaction = 0;
        min_interest = false;
    }
    else
    {
        if( ( T < Tm ) && ( I < I_max ) )
        {
            result += "Ідея не проходить поріг релізації і не є надзвичайно
цікавою!\n";
            index_reaction = 0;
        }
        else if( T >= Tm ) // завжди F < F_max
        {
            result += "Ідея проходить поріг релізації і не є надзвичайно
забороненою!\n";

            index_base = MEMORY_INTERESTING;
            index_reaction = memory_interest-
>getKnowledgeBase()[index_interest].getReaction();
            result += "Ситуація є більш цікавою, ніж забороненою!\n";
        }
        else //if( ( F >= F_max ) && ( I > I_max ) ) //завжди false, тому що
F < F_max
        {
            result += "Ситуація є одночасно надзвичайно цікавою і
надзвичайно забороненою, тому ситуація відноситься до області сумнівів!\n";
            index_reaction = 0;
        }
    }
}

```

```

    }
    else
    {
        result += "Ситуація відноситься до бази заборонених (" +
memory_forbidden->getKnowledgeBase()[index_forbidden].getSituation() +
        ", за пріоритетами " + situationPriorities( memory_forbidden-
>getKnowledgeBase()[index_forbidden].getSituation() ) +
        ") - можлива реакція " + reactions->getReaction(
memory_forbidden->getKnowledgeBase()[index_forbidden].getReaction() ) +
        "' - ідей свідомості!\n";

        if( I < I_min ) //завжди true
        {
            result += "Ідея не проходить поріг мінімальної інтересу ідеї!\n";
            index_reaction = 0;
            min_interest = false;
        }
    }

    result += "\nРеакція: " + getName() + " " + reactions->getReaction(
index_reaction ) + ".\n\n";

    //додавання до довгострокової пам'яті

    if( ( !doubtful ) && min_interest )
    {
        if( max_match_interest != 0 )
        {
            if( I > I_mem )
            {
                if( I >= I_plus )
                {
                    memory_interest->addKnowledge( Knowledge( situation,
index_reaction, I, F, S, QDateTime::currentDateTime().toMsecsSinceEpoch(), 0 ) );
                    result += "Ситуація є настільки цікавою, що вона додається
до бази цікавих ідей свідомості і її міра інтересу не змінюється із часом!\n";
                }
                else
                {
                    memory_interest->addKnowledge( Knowledge( situation,
index_reaction, I, F, S, QDateTime::currentDateTime().toMsecsSinceEpoch(), C ) );
                    result += "Ситуація є настільки цікавою, що вона додається
до бази цікавих ідей свідомості!\n";
                }
            }
        }
    }
}

```

```

    }
    else
    {
        result += "Ця ситуація вже є у базі цікавих ідей свідомості!\n";
    }

    if( max_match_forbidden != 0 )
    {
        if( F > F_mem )
        {
            if( F >= F_plus )
            {
                memory_forbidden->addKnowledge( Knowledge( situation,
index_reaction, I, F, S, QDateTime::currentDateTime().toMsecsSinceEpoch(), 0 ) );
                result += "Ситуація є настільки забороненою, що вона
додається до бази заборонених ідей свідомості і її міра протидії не змінюється із
часом!\n";
            }
            else
            {
                memory_forbidden->addKnowledge( Knowledge( situation,
index_reaction, I, F, S, QDateTime::currentDateTime().toMsecsSinceEpoch(), C ) );
                result += "Ситуація є настільки забороненою, що вона
додається до бази заборонених ідей свідомості!\n";
            }
        }
    }
    else
    {
        result += "Ця ситуація вже є у базі заборонених ідей центру
свідомості!\n";
    }
}
else
{
    if( I > F )
    {
        if( max_match_interest != 0 )
        {
            if( S > S_mem )
            {
                if( S >= S_plus )
                {

```

```

memory_interest->addKnowledge( Knowledge( situation,
index_reaction, I, F, S, QDateTime::currentDateTime().toMsecsSinceEpoch(), 0, true )
);
    result += "Сила ідеї є настільки високою, що вона
додається до бази цікавих ідей свідомості і її сила не змінюється із часом!\n";
    }
    else
    {
        memory_interest->addKnowledge( Knowledge( situation,
index_reaction, I, F, S, QDateTime::currentDateTime().toMsecsSinceEpoch(), C, true )
);
        result += "Сила ідеї є настільки високою, що вона
додається до бази цікавих ідей свідомості!\n";
    }
    }
    else
    {
        result += "Ця ситуація вже є у базі цікавих ідей центру
свідомості!\n";
    }
    }
    else
    {
        if( max_match_forbidden != 0 )
        {
            if( S > S_mem )
            {
                if( S >= S_plus )
                {
                    memory_forbidden->addKnowledge( Knowledge( situation,
index_reaction, I, F, S, QDateTime::currentDateTime().toMsecsSinceEpoch(), 0, true )
);
                    result += "Сила ідеї є настільки високою, що вона
додається до бази заборонених ідей свідомості і її сила не змінюється із часом!\n";
                }
                else
                {
                    memory_forbidden->addKnowledge( Knowledge( situation,
index_reaction, I, F, S, QDateTime::currentDateTime().toMsecsSinceEpoch(), C, true )
);
                    result += "Сила ідеї є настільки високою, що вона
додається до бази заборонених ідей свідомості!\n";
                }
            }
        }
    }
}

```

```

    }
    else
    {
        result += "Ця ситуація вже є у базі заборонених ідей центру
свідомості!\n";
    }
}
}

switch( index_base )
{
    case INSTINCT_INTERESTING:
    {
        result += "Рішення було знайдено у базі цікавих ідей
підсвідомості!\n";
        break;
    }
    case INSTINCT_FORBIDDEN:
    {
        result += "Рішення було знайдено у базі заборонених ідей
підсвідомості!\n";
        break;
    }
    case MEMORY_INTERESTING:
    {
        result += "Рішення було знайдено у базі цікавих ідей свідомості!\n";
        break;
    }
    case MEMORY_FORBIDDEN:
    {
        result += "Рішення було знайдено у базі заборонених ідей
свідомості!\n";
        break;
    }
    default:
    {
        result += "Рішення не було знайдено у жодній базі ідей центрів
свідомості і підсвідомості!\n";
        break;
    }
}

result += "\n-----\n";
result += "-----\n";

```

```

    delete reactions;
    return result;
}

```

```

//-----
Human::~~Human()
{
    delete instinct_interest;
    delete instinct_forbidden;
    delete memory_interest;
    delete memory_forbidden;
    priorities.clear();
}

```

A.11 Код файла mainwindow.h

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include "human.h"

#include <QMainWindow>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow( QWidget *parent = nullptr );
    ~MainWindow();

    void setParameters();

private:
    Ui::MainWindow *ui;
    std::vector<Human*> humans;

private slots:
    void On_listNames_currentIndexChanged( int index );

```

```
void On_btnAct_clicked();

};
#endif // MAINWINDOW_H
```

A.12 Код файлу mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "characteristics.h"

//-----
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    setWindowFlags( Qt::Window | Qt::MSWindowsFixedSizeDialogHint );
    setWindowTitle( QString( "Моделювання розумової діяльності людини" ) );
    setWindowIcon( QIcon( ":/img/app.png" ) );

    //зв'язування слотів
    connect( ui->listNames, SIGNAL( currentIndexChanged( int ) ), this, SLOT(
On_listNames_currentIndexChanged( int ) ) );
    connect( ui->btnAct, SIGNAL( released() ), this, SLOT(
On_btnAct_clicked() ) );

    //створення об'єктів Human

    //база цікавих ідей центру підсвідомості
    KnowledgeBase *base_instinct_interest = new KnowledgeBase();

    //база заборонених ідей центру підсвідомості
    KnowledgeBase *base_instinct_forbidden = new KnowledgeBase();

    //база цікавих ідей центру свідомості
    KnowledgeBase *base_memory_interest = new KnowledgeBase();

    //база заборонених ідей центру свідомості
    KnowledgeBase *base_memory_forbidden = new KnowledgeBase();
```

```
std::vector<unsigned int> priorities1 = { ITEM, CONSTITUTION, SEX,
SOCIAL_LEVEL, AGE, HAIR_COLOR };
```

```
//бідний високий чоловік із ножем
//реакція "втекти"
Knowledge *knowlegde1 = new Knowledge( "1xx220", 2 );
base_instinct_forbidden->addKnowledge( *knowlegde1 );
```

```
//бідний високий чоловік із ножем
//реакція "подзвонити в поліцію"
Knowledge *knowlegde2 = new Knowledge( "1xx220", 3 );
base_instinct_interest->addKnowledge( *knowlegde2 );
```

```
Human *human1 = new Human( "Іван (ситуація страху)",
*base_instinct_interest, *base_instinct_forbidden, *base_memory_interest,
*base_memory_forbidden,
0.4, 0.9, 0.4, 0.3, pow( 0.1, 6 ), 0.3, 0.4, 0.6, 0.4, 0.7, 0.65,
0.8, 0.8, 1.1, 0.9, 0.9, 1.5, priorities1 );
humans.push_back( human1 );
```

```
Human *human2 = new Human( "Петро (ситуація страху)",
*base_instinct_interest, *base_instinct_forbidden, *base_memory_interest,
*base_memory_forbidden,
0.9, 0.5, 0.4, 0.3, pow( 0.1, 6 ), 0.3, 0.4, 0.6, 0.4, 0.7, 0.65,
0.8, 0.8, 1.1, 0.9, 0.9, 1.5, priorities1 );
humans.push_back( human2 );
```

```
std::vector<unsigned int> priorities2 = { SEX, HAIR_COLOR,
SOCIAL_LEVEL, AGE, CONSTITUTION, ITEM };
```

```
//багата блондинка
//реакція "познайомитися"
Knowledge *knowlegde3 = new Knowledge( "00xxx2", 1, 0.99, 0, 0.392,
QDateTime::currentDateTime().toMsecsSinceEpoch(), 0 );
base_memory_interest->addKnowledge( *knowlegde3 );
```

```
//бідна блондинка
//реакція "відсутня"
Knowledge *knowlegde4 = new Knowledge( "00xxx0", 0, 0, 0.92, 0.392,
QDateTime::currentDateTime().toMsecsSinceEpoch(), pow( 0.1, 7 ) );
base_memory_forbidden->addKnowledge( *knowlegde4 );
```

```

        Human *human3 = new Human( "Сергій (ситуація сумнівів)",
        *base_instinct_interest,      *base_instinct_forbidden,      *base_memory_interest,
        *base_memory_forbidden,
        0.7, 0.6, 0.4, 0.3, pow( 0.1, 6 ), 0.3, 0.4, 0.6, 0.4, 0.7, 0.65,
        0.9, 0.9, 1.6, 0.98, 0.98, 1.8, priorities2 );
        humans.push_back( human3 );

        std::vector<unsigned int> priorities3 = { SEX, HAIR_COLOR,
        SOCIAL_LEVEL, AGE, CONSTITUTION, ITEM };

        base_memory_interest->deleteKnowledge( 0 );
        base_memory_forbidden->deleteKnowledge( 0 );

        //блондинка
        //реакція "відсутня"
        Knowledge *knowlegde5 = new Knowledge( "00xxxx", 0, 0, 0.98, 0.392,
        QDateTime::currentDateTime().toMsecsSinceEpoch(), 0 );
        base_memory_forbidden->addKnowledge( *knowlegde5 );

        //брюнетка
        //реакція "познайомитися"
        Knowledge *knowlegde6 = new Knowledge( "03xxxx", 1, 0.98, 0, 0.392,
        QDateTime::currentDateTime().toMsecsSinceEpoch(), 0 );
        base_memory_interest->addKnowledge( *knowlegde6 );

        Human *human4 = new Human( "Кирило (ситуація протидії)",
        *base_instinct_interest,      *base_instinct_forbidden,      *base_memory_interest,
        *base_memory_forbidden,
        0.8, 0.3, 0.4, 0.3, pow( 0.1, 6 ), 0.3, 0.4, 0.5, 0.4, 0.9, 0.9,
        0.95, 0.95, 1.6, 0.98, 0.98, 1.8, priorities3 );
        humans.push_back( human4 );

        //заповнення форми

        //заповнення параметрів людини
        QStringList nameList;
        for( unsigned int i = 0; i < humans.size(); i++ )
            nameList.append( QString::fromStdString( humans[i]->getName( true ) ) );

        ui->listNames->addItem( nameList );
        ui->listNames->setCurrentIndex( 0 );

```

```

setParameters();

//заповнення параметрів ситуації
QStringList nameSex = { "Жінка", "Чоловік" };
ui->listSex->addItems( nameSex );

QStringList nameHairColor = { "Блондин(ка)", "Рудий(а)", "Шатен(ка)",
"Брюнет(ка)" };
ui->listHairColor->addItems( nameHairColor );

QStringList nameAge = { "Менше 20", "20 - 30", "30 - 45", "45 - 60",
"Більше 60" };
ui->listAge->addItems( nameAge );

QStringList nameConstitution = { "Низький", "Середній", "Високий" };
ui->listConstitution->addItems( nameConstitution );

QStringList nameItem = { "Яблуко", "Сумка", "Ніж" };
ui->listItem->addItems( nameItem );

QStringList nameSocialLevel = { "Бідний", "Середнього достатку",
"Багатий" };
ui->listSocialLevel->addItems( nameSocialLevel );

}

//-----
void MainWindow::setParameters()
{
    ui->spba->setValue( humans[ui->listNames->currentIndex()->geta() );
    ui->spbb->setValue( humans[ui->listNames->currentIndex()->getb() );
    ui->spbc->setValue( humans[ui->listNames->currentIndex()->getc() );
    ui->spbd->setValue( humans[ui->listNames->currentIndex()->getd() );
    ui->spbC->setValue( humans[ui->listNames->currentIndex()->getC() * pow(
10, 6 ) );

    ui->spbTi_1->setValue( humans[ui->listNames->currentIndex()->getTi1() );
    ui->spbTi_2->setValue( humans[ui->listNames->currentIndex()->getTi2() );
    ui->spbTm->setValue( humans[ui->listNames->currentIndex()->getTm() );

    ui->spbI_min->setValue( humans[ui->listNames->currentIndex()->getImin()
);

```

```

        ui->spbI_max->setValue(          humans[ui->listNames->currentIndex()]-
>getImax() );
        ui->spbF_max->setValue(          humans[ui->listNames->currentIndex()]-
>getFmax() );

        ui->spbI_mem->setValue(          humans[ui->listNames->currentIndex()]-
>getImem() );
        ui->spbF_mem->setValue(          humans[ui->listNames->currentIndex()]-
>getFmem() );
        ui->spbS_mem->setValue(          humans[ui->listNames->currentIndex()]-
>getSmem() );

        ui->spbI_plus->setValue(         humans[ui->listNames->currentIndex()]-
>getIplus() );
        ui->spbF_plus->setValue(         humans[ui->listNames->currentIndex()]-
>getFplus() );
        ui->spbS_plus->setValue(         humans[ui->listNames->currentIndex()]-
>getSplus() );
    }

    //-----
    void MainWindow::On_listNames_currentIndexChanged( int index )
    {
        setParameters();
    }

    //-----
    void MainWindow::On_btnAct_clicked()
    {
        //оновлення параметрів людини
        humans[ui->listNames->currentIndex()]->seta( ui->spba->value() );
        humans[ui->listNames->currentIndex()]->setb( ui->spbb->value() );
        humans[ui->listNames->currentIndex()]->setc( ui->spbc->value() );
        humans[ui->listNames->currentIndex()]->setd( ui->spbd->value() );
        humans[ui->listNames->currentIndex()]->setC( ui->spbC->value() * pow(
0.1, 6 ) );

        humans[ui->listNames->currentIndex()]->setTi1( ui->spbTi_1->value() );
        humans[ui->listNames->currentIndex()]->setTi2( ui->spbTi_2->value() );
        humans[ui->listNames->currentIndex()]->setTm( ui->spbTm->value() );

        humans[ui->listNames->currentIndex()]->setImin( ui->spbI_min->value() );
        humans[ui->listNames->currentIndex()]->setImax( ui->spbI_max->value() );

```

```

        humans[ui->listNames->currentIndex()]->setFmax( ui->spbF_max->value()
    );

        humans[ui->listNames->currentIndex()]->setImem( ui->spbI_mem->value()
    );
        humans[ui->listNames->currentIndex()]->setFmem( ui->spbF_mem->value()
    );
        humans[ui->listNames->currentIndex()]->setSmem( ui->spbS_mem->value()
    );

        humans[ui->listNames->currentIndex()]->setImem( ui->spbI_mem->value()
    );
        humans[ui->listNames->currentIndex()]->setFmem( ui->spbF_mem->value()
    );
        humans[ui->listNames->currentIndex()]->setSmem( ui->spbS_mem->value()
    );

    //зчитування ситуації
    QString situation;
    situation += QString::number( ui->listSex->currentIndex() );
    situation += QString::number( ui->listHairColor->currentIndex() );
    situation += QString::number( ui->listAge->currentIndex() );
    situation += QString::number( ui->listConstitution->currentIndex() );
    situation += QString::number( ui->listItem->currentIndex() );
    situation += QString::number( ui->listSocialLevel->currentIndex() );

    ui->labelCodeValue->setText( situation );

    //результат
    ui->textLog->append( QString( "Людина: %1." ).arg(
    QString::fromStdString( humans[ui->listNames->currentIndex()]->getName() ) ) );
    ui->textLog->append( QString( "Пріоритети розташовані: %1." ).arg(
    QString::fromStdString( humans[ui->listNames->currentIndex()]->prioritiesToString()
    ) ) );
    ui->textLog->append( QString::fromStdString( humans[ui->listNames->
    >currentIndex()]->result( situation.toStdString() ) ) );

    }

    //-----
    MainWindow::~MainWindow()
    {
        humans.clear();
        delete ui;
    }

```

```
}
```

A.13 Код файла main.cpp

```
#include "mainwindow.h"
```

```
#include <QApplication>
```

```
int main(int argc, char *argv[])  
{  
    QApplication a(argc, argv);  
    MainWindow w;  
    w.show();  
    return a.exec();  
}
```

ДОДАТОК Б
Слайди презентації

НУ “Запорізька політехніка”
Кафедра програмних засобів
Дипломна кваліфікаційна робота бакалавра



Програмне забезпечення для моделювання розумової діяльності людини

Виконала:
ст. групи КНТ-129

Вероніка КАРНАУХ

Керівник:
к.т.н., доцент

Сергій СЕРДЮК

Рисунок Б.1 – Слайд №1

Мета розробки

- Мета роботи – розробка програмного забезпечення, що дозволить моделювати мислячу систему людини на основі р-адичних дерев.
- Об’єкт дослідження – процес мислення людини.
- Предмет дослідження – програмне забезпечення для моделювання розумової діяльності людини.

2/15

Рисунок Б.2 – Слайд №2

Задачі, що вирішуються

- проведення аналізу предметної області;
- розробка технічного завдання;
- розробка структури програмного забезпечення;
- моделювання програмного забезпечення;
- розробка класів і реалізація програмного забезпечення;
- оформлення документації.

3/15

Рисунок Б.3 – Слайд №3

Аналіз мов програмування

	C	C++	Java	Python
Компільована/інтерпретована	К	К	К	І
Мультипарадигмальність	+	++	+++	+++
Безпечність типізації	-	-	+	+
Вивід типів	-	+-	-	+
Стандартизованість	+	+	-	-
Ефективність	+++	++	++	+

4/15

Рисунок Б.4 – Слайд №4

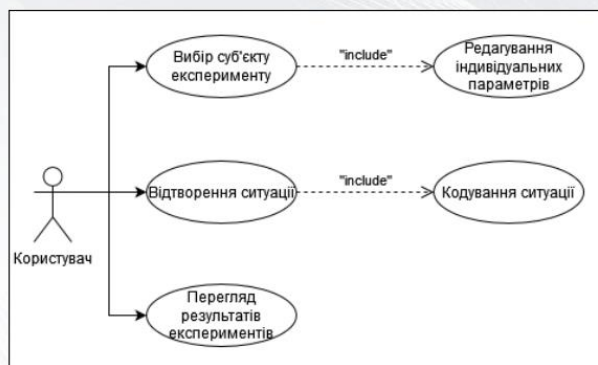
Аналіз середовищ розробки

	Microsoft Visual Studio	NetBeans	Qt Creator
Багатоплатформність	+ -	+	+
Бібліотеки інтерфейсу	+	-	+
Статичний аналіз коду	+	-	+
Безкоштовність	+ -	+	+
Підтримка репозиторіїв	++	+	++

5/15

Рисунок Б.5 – Слайд №5

Діаграма прецедентів



6/15

Рисунок Б.6 – Слайд №6

Діаграма послідовностей



7/15

Рисунок Б.7 – Слайд №7

Основні формули методу моделювання

$$p_m(x, y) = \frac{1}{m^k}$$

$$I(J) = \frac{1}{1 + p(J, D_i)}$$

$$T(J) = a * I(J) - b * F(J)$$

$$F(J) = \frac{1}{1 + p(J, D_f)}$$

$$S(J) = c * I(J) + d * F(J)$$

$$l_{mem}(t) = e^{-Ct}$$

$$I(t, J) = l_{mem}(t - t_0) * I(J)$$

$$F(t, J) = l_{mem}(t - t_0) * F(J)$$

$$S(t, J) = l_{mem}(t - t_0) * S(J)$$

8/15

Рисунок Б.8 – Слайд №8

Приклад кодування ситуацій

Код	Стать	Колір волосся	Вік	Зріст	Предмет	Соціальний рівень
0	Жінка	Блондин(ка)	Менше 20	Низький	Яблуко	Бідний
1	Чоловік	Рудий(а)	20 - 30	Середній	Сумка	Середнього достатку
2		Шатен(ка)	30 - 45	Високий	Ніж	Багатий
3		Брюнет(ка)	45 - 60			
4			Більше 60			

9/15

Рисунок Б.9 – Слайд №9

Графічний інтерфейс

Моделювання розумової діяльності людини

Редактор ситуації:

Стать: Жінка | Колір волосся: Блондин(ка)

Вік: Менше 20 | Зріст: Низький

Предмет: Яблуко | Соціальний рівень: Бідний

Хід моделювання

Індивідуальні параметри людини

a	b	c	d	C, 10 ⁻⁶
0,40	0,90	0,40	0,30	1,00

T1_1: 0,300 | I_min: 0,400 | I_mem: 0,800 | I+: 0,900

T1_2: 0,400 | I_max: 0,700 | F_mem: 0,800 | F+: 0,900

Tm: 0,600 | F_max: 0,650 | S_mem: 1,100 | S+: 1,500

Код ситуації: 000000

Почати

10/15

Рисунок Б.10 – Слайд №10

Моделювання ситуації “Страх”

Редактор ситуації:

Стать

Чоловік

Колір волосся

Блондин(ка)

Вік

Менше 20

Зріст

Високий

Предмет

Нік

Соціальний рівень

Бідний

Хід моделювання

$I = 1.0000$ $F = 1.0000$ $T = -0.5000$

Ситуація одночасно відноситься до бази знань (1x220, за пріоритетами 2210x) - можлива реакція 'злякався і викликав пожежу' - та бази заборонених (1x220, за пріоритетами 2210x) - можлива реакція 'злякався і втік' - цієї підсвідомості!

Ситуація є більш забороненою, ніж чаканою!

Ідея пролонгує поріг угодженості!

Ідея пролонгує поріг фізичної реалізації!

Решення: Іван злякався і втік.

Рішення було знайдено у базі заборонених (цей підсвідомості!)

$I = 1.0000$ $F = 1.0000$ $T = 0.4000$

Ситуація одночасно відноситься до бази знань (1x220, за пріоритетами 2210x) - можлива реакція 'злякався і викликав пожежу' - та бази заборонених (1x220, за пріоритетами 2210x) - можлива реакція 'злякався і втік' - цієї підсвідомості!

Ситуація є більш чаканою, ніж забороненою!

Ідея пролонгує поріг угодженості!

Ідея пролонгує поріг фізичної реалізації!

Решення: Петро злякався і викликав пожежу.

Рішення було знайдено у базі знань (цей підсвідомості!)

Людина

Петро (ситуація страху)

Індивідуальні параметри людини

a	b	c	d	$C, 10^{-6}$
0,90	0,50	0,40	0,30	1,00

Ti_1	I_min	I_mem	I+
0,300	0,400	0,800	0,900

Ti_2	I_max	F_mem	F+
0,400	0,700	0,800	0,900

Tm	F_max	S_mem	S+
0,600	0,650	1,100	1,500

Код ситуації

100220

11/15

Рисунок Б.11 – Слайд №11

Моделювання ситуації “Сумніви”

Редактор ситуації:

Стать

Жінка

Колір волосся

Блондин(ка)

Вік

Менше 20

Зріст

Низький

Предмет

Яблучо

Соціальний рівень

Бідний

Хід моделювання

Вибивається пошук у свідомості!

$I = 0.9730$ $F = 1.0000$ $T = 0.0811$ $S = 0.6892$

Ситуація одночасно відноситься до бази знань (00x2, за пріоритетами 0020x) - можлива реакція 'злякався і викликав пожежу' і підійшов позитивно! - та бази заборонених (00x2, за пріоритетами 0020x) - можлива реакція 'нік не вирізував на ситуацію' - цієї свідомості!

Ситуація є одночасно надзвичайно чаканою і надзвичайно забороненою, тому ситуація відноситься до області сумнівів!

Решення: Сергій сумнівається і не може обрати рішення.

Рішення не було знайдено у жодній базі (цей центр свідомості і підсвідомості!)

Міра протидії 00x2 у базі заборонених (цей свідомості зменшилася із часом, тому її було відкинуто!)

$I = 0.9730$ $F = 0.0000$ $T = 0.6811$ $S = 0.3892$

Ідея пролонгує поріг реалізації і не є надзвичайно забороненою!

Решення: Сергій задивляється тихо похитуючи і підійшов позитивно!

Рішення було знайдено у базі знань (цей свідомості!)

Людина

Сергій (ситуація сумнівів)

Індивідуальні параметри людини

a	b	c	d	$C, 10^{-6}$
0,70	0,60	0,40	0,30	1,00

Ti_1	I_min	I_mem	I+
0,300	0,400	0,900	0,980

Ti_2	I_max	F_mem	F+
0,400	0,700	0,900	0,980

Tm	F_max	S_mem	S+
0,600	0,650	1,600	1,800

Код ситуації

000000

12/15

Рисунок Б.12 – Слайд №12

Моделювання ситуації “Протидія”

[illegible]

13/15

Рисунок Б.13 – Слайд №13

Висновки

- Було створено програмне забезпечення для моделювання розумової діяльності людини на базі обраної математичної моделі мозку, проведено аналіз предметної області, спроєктовано архітектуру та моделі програмного забезпечення, оформлено необхідну документацію.
- Розроблена система відповідає вимогам щодо інформаційної повноти, програмної сумісності та надійності.
- Практична значущість роботи полягає у можливості дослідження закономірностей процесів мислення та імітації розумової діяльності людини за допомогою зручного програмного забезпечення, що дозволяє експериментувати з індивідуальними психофізіологічними параметрами людини.
- Результати цієї роботи були представлені на науково-практичній конференції "Тиждень науки-2023" [1].

14/15

Рисунок Б.14 – Слайд №14

Перелік публікацій

1. Карнаух, В. Д. Моделювання розумової діяльності людини [Текст] / В. Д. Карнаух // Тези доповідей науково-практичної конференції, Запоріжжя, 24–28 квітня 2023 р. : Тижень науки-2023, Запоріжжя, 24–28 квітня 2023 р. : тези доповідей. – Запоріжжя: НУ "Запорізька політехніка", 2023. У друці.

15/15

Рисунок Б.15 – Слайд №15