

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ.....	10
ВСТУП.....	11
РОЗДІЛ 1 Опис предметної області.....	13
1.1 Пам'ять і її тренування.....	13
1.2 Метод запам'ятовування Цицерона	
Висновок до розділу 1.....	19
РОЗДІЛ 2 СИСТЕМНИЙ АНАЛІЗ ТА АНАЛІЗ ВИМОГ.....	20
2.1. Найменування ПП.....	20
2.2. Призначення і область застосування.....	20
2.3. Вимоги до ПП.....	20
2.3.1. Вимоги до функціональних характеристик.....	20
2.3.2 Вимоги до надійності.....	21
2.3.3 Час відновлення після відмови.....	21
2.3.4. Відмови через некоректні дії користувачів системи.....	21
2.4. Умови експлуатації.....	22
2.4.1. Кліматичні умови експлуатації.....	22
2.4.2. Вимоги до кваліфікації та чисельності персоналу.....	22
2.4.3. Вимоги до інформаційної та програмної сумісності.....	22
2.4.4. Спеціальні вимоги.....	23
2.5. Вимоги до програмної документації.....	23
2.6. Техніко-економічні показники.....	24
2.6.1. Економічні переваги розробки.....	24

ЛД21.9102.1300.1732.ПЗ

Зм. Лист	№ докум.	Підп.	Дата	Додаток для пам'яті «Метод Цицерона»		
Розроб.	Босенко Н.В.			Лист	Листів	
Перев.	Яценко О.В.			7		
Рецен.	Овчаренко Г.Р.			/ «КПІ»		
Н. контр.	Кисляк С.В.			. БМК		
Затв.				р. ЛД-21		

2.7. Стадії та етапи розробки.....	24
2.7.1. Стадії розробки.....	24
2.7.2. Етапи розробки.....	25
2.8 Модель ЖЦ.....	26
Висновки до розділу 2.....	26
РОЗДІЛ 3: ПРОЕКТУВАННЯ.....	27
3.1. Діаграма функціонального моделювання.....	27
3.4. Use-case diagram.....	27
3.5 Sequence diagram.....	28
Висновок до розділу 3.....	28
РОЗДІЛ 4 ОСНОВНІ ОСОБЛИВОСТІ РОЗРОБКИ ДОДАТКУ.....	29
4.1 Мова програмування C Sharp.....	29
4.2 Мова програмування SQL.....	32
4.2.1. Переваги.....	36
4.2.2 Недоліки.....	37
4.3 Середовище розробки.....	38
РОЗДІЛ 5 КЕРІВНИЦТВО КОРИСТУВАЧА.....	42
5.1 Як встановити ПП.....	42
5.2 Як зробити резервні копії БД.....	42
5.3. Інструкція для користувача.....	42
Висновки до розділу 5.....	47
РОЗДІЛ 6 ОХОРОНА ПРАЦІ.....	48
Вступ.....	48
6.1 Аналіз умов на робочому місці.....	48

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуц

8

6.1.1 Характеристика приміщення.....	48
6.2 Оцінка небезпечних та шкідливих виробничих факторів.....	51
6.3 Мікроклімат.....	52
6.4 Освітлення.....	53
6.5 Шум.....	55
6.6 Пожежна безпека.....	56
Висновки до розділу 6.....	57
ЗАГАЛЬНІ ВИСНОВКИ.....	58
ПЕРЕЛІК ПОСИЛАНЬ.....	59
Додаток А.....	61

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

9

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

ОС – операційна система

ПП – програмний продукт

SQL (Structured query language) - мова структурованих запитів

СКБД – Система керування базами даних

ПК – персональний комп'ютер

CLR (Common Language Runtime) - це компонент пакету Microsoft .NET Framework, віртуальна машина, на якій виконуються всі мови платформи .NET Framework.

ДСанПіН – Державні Санітарні Правила і Норми

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

10

ВСТУП

Пам'ять — психічний процес, який полягає в закріпленні, збереженні та наступному відтворенні минулого досвіду, дає можливість його повторного застосування в життєдіяльності людини.

Неможливо отримати хорошу пам'ять не тренуючи її . Гарна пам'ять - це навичка , яка потребує тренування і витративши якийсь час на виконання вправ щодо поліпшення пам'яті , ви зможете набагато краще запам'ятовувати. Розвиток пам'яті, уваги та мислення є суміжними завданнями.

На сьогоднішній день створено дуже багато додатків, спрямовані на тренування пам'яті і здатності цілеспрямованого запам'ятовування матеріалу.

Запам'ятовування - це процес пам'яті , за допомогою якого відбувається сприйняття нової інформації та запис цієї інформації в загальну систему мислення та асоціативних зв'язків . Ключовою функцією запам'ятовування є створення смислових зв'язків , як результату роботи нашого мислення і інтелекту над змістом матеріалу, що запам'ятовується .

У зв'язку з величезним потоком інформації , який нам доводиться щодня пропускати через себе , наш мозок потребує періодичного стресу . Існує велика кількість вправ , виконання яких дозволить вам поліпшити можливості вашої пам'яті в запам'ятовуванні .

Метод Цицерона - це унікальний мнемонічний прийом, якому може навчитися кожен. Інформація, що надходить у мозок людини, краще запам'ятається, якщо встановлено зв'язок між подіями.

Метою дипломного проектування є створення додатку для пам'яті «Метод Цицерона».

Для досягнення поставленої мети необхідно виконати наступні завдання:

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

11

- Оглянути та проаналізувати створенні сучасні аналоги програмних продуктів та web-ресурсів, що використовуються для тренування пам'яті.
- Провести вибір мови програмування та платформи Windows та на базі даних кроків розробити програмний продукт.

Об'єктом даного проекту є вправи для підвищення ефективності тренування пам'яті користувача додатку.

Предметом дослідження являється система реєстрації та контролю покращення запам'ятовування користувача додатку.

На сьогоднішній час тема дослідження дипломного проекту актуальна, оскільки створення додатку, за допомогою якого будуть використовуватись вправи на запам'ятовування, де встановлюються смислові зв'язки між двома явищами, попереднє визначення, якою може бути смисловий зв'язок між цими явищами, подіями або діями, сприяє більш міцному запам'ятовуванню. Асоціативні зв'язки, навіть коли вони абсолютно неймовірні за змістом, то вони запам'ятовуються надовго.

Новизна роботи полягає у тому, що додаток «Метод Цицерона» раніше не був реалізований у вигляді ПП. Програма розрахована для використання за ПК без підключення Інтернету, що є зручним для багатьох людей.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

12

РОЗДІЛ 1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Пам'ять і її тренування

Пам'ять - здатність нервової системи сприймати і зберігати інформацію і витягувати її для вирішення різноманітних завдань і побудови своєї поведінки. Завдяки цій складній і важливій функції мозку людина може накопичувати досвід і використовувати його в майбутньому.

Інформаційні сигнали спочатку впливають на аналізатори, викликаючи в них зміни, які тримаються, як правило, не більше 0,5 секунди. Ці зміни називають сенсорної пам'яттю - вона дозволяє людині зберігати, наприклад, зоровий образ під час миготіння або переглядати кінофільм, сприймаючи єдність зображення, незважаючи на змінялися кадри. У процесі тренування тривалість дії цього виду пам'яті може подовжуватися до десятків хвилин - в цьому випадку говорять про ейдетичеської пам'яті, коли її характер стає підконтрольним свідомості (принаймні, частково). Наступною за сенсорної пам'яттю за тривалістю зберігання інформації виділяють короткострокову пам'ять, яка дозволяє оперувати інформацією десятки секунд. Найбільш ж важлива, найбільш значуща частина інформації зберігатися в довготривалій пам'яті, яка забезпечує ці функції роки і десятиліття.

Для того щоб запам'ятати інформацію, перш за все необхідно сконцентрувати на ній увагу, після чого зняти зайву напругу, що заважає запам'ятовуванню. З цією метою необхідно навчитися розслаблятися (за допомогою аутотренінгу, цілеспрямованого довільного розслаблення окремих м'язових груп, особливо рук, і т.д.). Тренування самонавіювання, образно-чуттєвого мислення, уваги спрощує використання раціональних мнемотехніческих прийомів. Найпростішим з них є метод асоціацій:

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

13

наприклад, якщо потрібно запам'ятати якісь нові слова, їх пов'язують з добре відомими словами або з образними асоціаціями. Як показує практика, чим неймовірнішою або навіть абсурднішою асоціацією, тим краще вони запам'ятовуються.

Інформацію, яку потрібно запам'ятати, повторюють через якийсь час, причому інтервал між повтореннями повинен становити не менше 1 хвилини. Водночас оптимальний інтервал повторення в залежності від складності та обсягу інформації, а також індивідуальних особливостей людини коливається від 10 хвилин до 16 годин.

Ідеально, якщо останнє повторення здійснюється перед сном - це підвищує якість запам'ятовування. Мабуть, опрацювання матеріалу перед сном взагалі сприяє його кращому запам'ятовуванню (це обумовлено тим, що переробка інформації у сні відбувається в зворотній послідовності, тобто спочатку переробляється остання, найсвіжіша).

При запам'ятовуванні необхідно по можливості максимально задіяти всі механізми мозку. Наприклад, при вивченні усного матеріалу бажано не тільки вимовляти слова вголос, але й уважно їх прочитувати, наговорювати їх на магнітофон з подальшим прослуховуванням, записувати на папері основні положення нового матеріалу, слова, дати і т.д. Завдяки цьому, активуються багато аналізаторів, пов'язані з різними областями кори мозку. Оскільки процес пам'яті - це робота всього мозку (точніше, навіть всього організму), то така його активація надзвичайно сприятливо позначається на якості запам'ятовування.

Природно, що при виборі оптимального варіанту мнемотехніки (тобто способу запам'ятовування) необхідно пам'ятати про індивідуальні особливості людини, переважній вигляді пам'яті, особливості запам'ятовування, рівні мотивації і т.д.

Регулярні тренування пам'яті, в тому числі повторення потрібного матеріалу, підвищують можливості запам'ятовування. Погіршення якості

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

14

запам'ятовування може свідчити про недостатність тренувань, про високий рівень напруги, тривожності, втомі і вимагає аналізу або самоаналізу для виправлення становища.

У реалізації пам'яті роль усвідомлюваного і неусвідомлюваного безперечна, хоча ступінь їх відносин в цьому процесі досить складна для опису. Слід зазначити, що свідоме запам'ятовування інформації володіє відносно невеликий інформаційною ємністю, а область неусвідомлюваного - колосальною, практично безмежною. Можливості неусвідомлюваного проявляють себе, зокрема, у сновидіннях людини, де виявляється, що мозок може запам'ятовувати все, в тому числі і, здавалося б, зовсім непотрібні деталі. Є підстави вважати, що ці можливості мозку можна за цілеспрямованої тренуванні та спеціальної організації частково використовувати для довільного запам'ятовування. Можуть допомогти цьому різні психотехніки, про які йшлося вище, - вони дозволяють активувати підсвідомість, змінити звичні співвідношення між свідомістю та несвідомими і розкрити можливості людини.

Правила запам'ятовування (навчання). Для хороших результатів у галузі тренування пам'яті, крім зазначених раніше умов, необхідно обов'язково враховувати ще ряд положень. По суті - це психофізіологічні основи успішного навчання, тісно пов'язані з правилами утворення умовних рефлексів. Для успішної тренування пам'яті та запам'ятовування необхідно

- володіти основними знаннями, необхідними для розуміння інформації;
- усвідомлювати свою мету;
- проявляти максимальний інтерес до інформації, бажання її запам'ятати;
- створювати або вибирати сприятливі умови для роботи;
- бути в гарному психофізіологічному стані;
- сконцентрувати увагу на потрібній інформації, виключити причини неуважності;
- регулярно тренувати свою пам'ять і всі її компоненти, використовувати всі механізми, можливості психіки для поліпшення пам'яті.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

15

1.2 Метод запам'ятовування Цицерона

Метод запам'ятовування Цицерона носить ім'я одного з великих римських політиків, який прославився тим, що в процесі своїх яскравих виступів ніколи не користувався записами. При цьому оратор оперував безліччю цифр та імен, фактів, а також досить часто використовував цитати.

Відповідно до деяких джерел, метод Цицерона вельми успішно застосовували задовго до офіційного його автора. Зокрема, відомий давньогрецький поет Симонид активно і успішно його практикував.

Цей метод простий і одночасно ефективний. Інша його назва - система римської кімнати. Він ґрунтується на тому, що ключові одиниці, які треба запам'ятати, необхідно розставляти в думках в певному порядку, представляючи в голові звичне приміщення. Після такої процедури достатньо лише відновити в пам'яті цю кімнату, щоб відновити все, що ви фіксували в пам'яті. Цицерон, коли був зайнятий підготовкою до публічного виступу, походжав по дому і розміщував в голові основні ключові моменти своєї промови в різних місцях.

Перед тим, як почати практикувати запам'ятовування по методу Цицерона, визначтеся, яким чином ви плануєте обходити кімнату. Виберіть послідовність місць, в яких ви плануєте подумки розміщувати ключові одиниці. Для деяких уявлення кімнати в голові буде достатньо. Однак у перший раз не буде зайвим пройтися по будинку, як це робив великий оратор. При цьому варто вибирати звичну кімнату, наприклад - свій особистий кабінет. Можете віддати перевагу способу пересування за годинниковою стрілкою.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

16

З досвідом ви будете в змозі задіяти все більше число предметів і місць, тим самим розширюючи свої можливості для фіксації інформації. Ви можете використовувати абсолютно будь-які предмети. Нехай це буде диван, телевізор, поличка, робочий стіл, комп'ютер і так далі. Подумки переміщатися можна і зліва направо, і вгору-вниз. Вам не обов'язково обмежуватися тільки своєю кімнатою. Для більш ефективної фіксації в пам'яті можна використовувати знайомі маршрути і т.п.

Вам необхідно пов'язувати ключові моменти інформації, яку ви намагаєтеся запам'ятати, з певними предметами за допомогою асоціацій. Починаючи практикувати даний метод, не буде зайвим мати в голові набір елементів кімнати. Таким чином, у вас завжди буде «підручний» набір так званих ментальних гачків, за які ви зможете зачепити потрібні ключові моменти.

В якості прикладу розглянемо, як можна легко запам'ятати список наступних елементів: велосипед, рюкзак, пляшка, пластилин. Ці елементи будуть розміщені послідовно у відповідності зі схемою будинку. Почнемо ми з коридору. Необхідно використовувати нестандартні зв'язки, щоб його відтворення не склало труднощів навіть через довгий час.

Велосипед розміщуємо на вішалці біля дверей, причому знаходиться він в перевернутому положенні. Така асоціація легко відкладеться в нашій голові. Далі йде рюкзак, який висить на дверях, а з нього визирає кіт. У пляшку ми ставимо квітку соняшника і розміщуємо його на тумбочці. Дзеркало, розташоване в передпокої, повністю заклеюємо пластиліном, тим самим створюючи нестандартну асоціацію і фіксуючи останній ключовий елемент в нашому списку. Таким чином, у нас вийшов досить незвичайний асоціативний ряд, який ми легко зможемо відновити в голові.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

17

Дуже важливо попередньо визначити ті предмети, що будуть зустрічатися нам на шляху. В іншому випадку є ризик того, що ви будете самі шукати деталі, які буде легше пов'язати з певним ключовим словом. У підсумку, запам'ятовування буде неефективним.

Зрозуміло, метод Цицерона варто застосовувати для задач значно складніше. Вище описаний приклад лише допомагає зрозуміти, як відбувається запам'ятовування інформації. Даний метод продуктивний в процесі фіксації в пам'яті тексту, плану на день, порядку необхідних телефонних дзвінків та іншого. Більше того, коли інформація являє собою логічно пов'язані дані, а не просто набір слів, ви зможете неодноразово використовувати одну кімнату. При цьому ряди ключових елементів не переміщаються, а ви будете в змозі легко відновлювати дані по конкретній темі.

Щоб навчитися використовувати даний метод, достатньо кількох тренувань. Це і є його головним плюсом, якщо порівнювати дану методику з іншими. Більше того, цю техніку можна використовувати в будь-якому місці. А те приміщення, де ви знаходитесь, може стати чудовою відправною точкою для запам'ятовування інформації. Вам немає необхідності використовувати метод пов'язаних асоціацій або довго відновлювати ланцюжок, як це потрібно при застосуванні методу послідовних асоціацій. Вам лише потрібно відновити в пам'яті знайоме приміщення або використовувати те, де ви знаходитесь в конкретний момент. Цілком достатньо розглядати кімнату, при цьому розставляючи ключові слова і прив'язуючи їх до якихось предметів. Відновити інформацію також просто. Досить згадати обстановку. Застосовуючи цей метод, ви здивуєтеся, наскільки легко і швидко можна засвоїти навіть вельми складний матеріал.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

18

Висновок до розділу 1

В даному розділі було запропоновані методи тренування пам'яті, проаналізовано що таке «Метод Цицерона».

Ми отримали основні задачі для створення корисного програмного продукту для пам'яті. Додаток буде не лише виконувати свої безпосередні функції, а й записували інформацію про користувача та візуально відображати зміни.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

19

РОЗДІЛ 2 СИСТЕМНИЙ АНАЛІЗ ТА АНАЛІЗ ВИМОГ

2.1. Найменування ПП

Найменування ПП: «Додаток для пам'яті «Метод Цицерона».

2.2. Призначення і область застосування

ПП призначений для:

- 1) Створення додатку для використання в різних закладах;
- 2) ПП має власний інтерфейс і може використовуватись для:
 - проходження вправ для покращення запам'ятовування.
 - збереження результатів тренувань.

2.3. Вимоги до ПП

2.3.1. Вимоги до функціональних характеристик

ПП повинен забезпечувати можливість виконання перерахованих нижче функцій:

- 1) виконувати призначення що були вказані в п. 2.
- 2) видавати статистику проходження вправ.
- 3) працювати на найбільш відомих операційних системах (Windows, Mac OS, Linux).

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

20

2.3.2 Вимоги до надійності

Надійне (стійке) функціонування ПП має бути забезпечене виконанням Замовником сукупності організаційно-технічних заходів, перелік яких наведено нижче:

- а) організацією безперебійного живлення технічних засобів;
- б) відсутністю сторонніх або шкідливих програм, що можуть привести до непрацездатності даної програми

2.3.3 Час відновлення після відмови

Час відновлення після відмови, викликаного збоєм електроживлення технічних засобів (іншими зовнішніми чинниками), не фатальним збоєм (не крахом) операційної системи, не повинно перевищувати 10-ти хвилин за умови дотримання умов експлуатації технічних і програмних засобів.

Час відновлення після відмови, викликаного несправністю технічних засобів, фатальним збоєм (крахом) операційної системи, не повинно перевищувати часу, необхідного на усунення несправностей технічних засобів і переустановлення програмних засобів.

2.3.4. Відмови через некоректні дії користувачів системи

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Відмови ПП унаслідок некоректних дій користувача при взаємодії з ПП через інтерфейс неприпустимі. У ПП необхідна реалізація засобів захисту від подання некоректних даних.

2.4. Умови експлуатації

2.4.1. Кліматичні умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні задовольняти вимогам, що пред'являються до технічних засобів в частині умов їх експлуатації.

Нормальні кліматичні умови експлуатації комп'ютера (або ноутбука): температура повітря $(20\pm 5)^{\circ}\text{C}$, відносна вологість до $(60\pm 15)\%$, атмосферний тиск від 84 кПа до 107 кПа.

2.4.2. Вимоги до кваліфікації та чисельності персоналу

Для роботи з ПП достатньо однієї людини з середнім рівнем комп'ютерної грамотності.

2.4.3. Вимоги до інформаційної та програмної сумісності

2.4.3.1. Вимоги до інформаційних структур і методів розв'язання

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

22

Додаткові вимоги не пред'являються

2.4.3.2. Вимоги до вихідних кодів та мов програмування

Додаткові вимоги не пред'являються.

2.4.3.3. Вимоги до програмних засобів, які використовуються

ПП

Системні програмні засоби, що використовуються ПП, повинні бути представлені ліцензійною локалізованою версією операційної системи Windows XP (або наступні версії)

2.4.3.4. Вимоги до захисту інформації та ПП

Вимоги до захисту інформації та ПП не пред'являються.

2.4.4. Спеціальні вимоги

ПП повинен забезпечувати роботу одного користувача за допомогою графічного інтерфейсу користувача.

2.5. Вимоги до програмної документації

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

23

Склад програмної документації повинен включати в себе:

- 1) Документацію етапу системного аналізу та аналізу вимог;
- 2) Документацію етапу проектування;
- 3) Документацію етапу кодування;
- 4) Документацію етапу тестування;
- 5) Інструкція для користувача;
- 6) Підсумки роботи.

2.6. Техніко-економічні показники

2.6.1. Економічні переваги розробки

Орієнтовна економічна ефективність не розраховуються. Аналогія не проводиться, зважаючи на унікальності пропонованих вимог до розробки.

2.7. Стадії та етапи розробки

2.7.1. Стадії розробки

Розробка повинна бути проведена в 5 стадій:

- 1 системний аналіз та аналіз вимог;
- 2 проектування;
- 3 створення програми:
 - а) проектування(створення UML діаграми)
 - б) розробка інтерфейсу по технології WPF(кодування)тестування і налагодження роботи ПП;

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

24

- 4 розробка документації;
- 5 оформлення розрахункової роботи.

2.7.2. Етапи розробки

На стадії системного аналізу та аналізу вимог повинно бути сформоване технічне завдання, діаграма Ганта та мережевий графік, які узгоджені та затверджені Заказником.

На стадії проектування повинні бути виконані перераховані нижче етапи робіт:

- 1 розробка Ієрархічної структури та розрахунок нев'язки;
- 2 розробка ERD. SADT. DFD діаграм;
- 3 розробка блок-схем основної частини ПП та усіх її допоміжних функцій;
- 4 розробка USE-CASE діаграми за допомогою мови UML.

На стадії кодування повинні бути виконані перераховані нижче етапи робіт:

- 1) реалізація спільного інтерфейсу ПП в VS;
- 2) реалізація модулів ПП в середовищі MySQL.

На стадії тестування і налагодження роботи ПП проводиться випробування системи на наявність помилок, достовірність роботи всіх підсистем і системи в цілому, оцінка простоти та інформативності інтерфейсу користувача.

На стадії розробки документації повинні бути виконані перераховані нижче етапи робіт:

- 1 оформлення документації, яка перерахована на етапі системного аналізу та аналізу вимог;
- 2 оформлення усіх результатів діяльності (діаграм та схем) на етапі проектування;
- 3 оформлення результатів етапів кодування та тестування;
- 4 оформлення посібника користувача.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

25

На стадії оформлення розрахункової роботи проводиться підведення підсумків та опис результатів усіх виконаних робіт.

2.8 Модель ЖЦ

Для проектування ПП було обрано каскадну модель ЖЦ, в якій перехід на наступний етап означає повне завершення робіт на попередньому етапі.

Я вважаю, що ця модель в порівнянні з іншими є доцільною, так як етапи робіт виконуються в логічній послідовності, що дозволяє планувати терміни завершення всіх робіт і відповідні затрати. Кожний етап завершується випуском повної документації. Також можна більш точно сформулювати необхідні вимоги на початку проектування, особливо з технічної точки зору.

Обрана модель дає змогу написати одну з найбільш детальних документацій до кожного етапу.

Так як я студентка бакалавру, то по закінченню роботи, в мене не буде можливості брати участь в наступній ітерації.

Висновки до розділу 2

В ході системного аналізу та аналізу вимог було виявлено основні вимоги до функціонування ПП. Визначившись з вимогами до ПП, можна розпочинати проектування ПП.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

26

РОЗДІЛ 3: ПРОЕКТУВАННЯ

3.1. Діаграма функціонального моделювання

SADT діаграма, згідно до завдань, що були сформульовані на етапі системного аналізу та аналізу вимог, показана на рисунку 3.1.

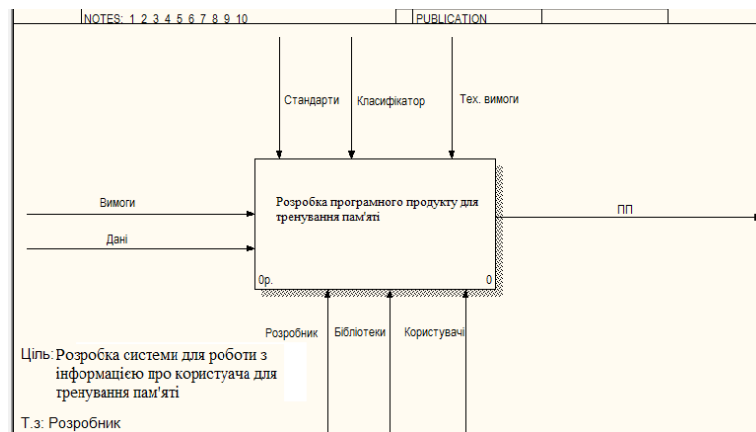


Рис. 3.1

3.4. Use-case diagram

Діаграма USE – CASE, згідно до завдань, що були сформульовані на етапі системного аналізу та аналізу вимог, показана на рисунку 3.2.

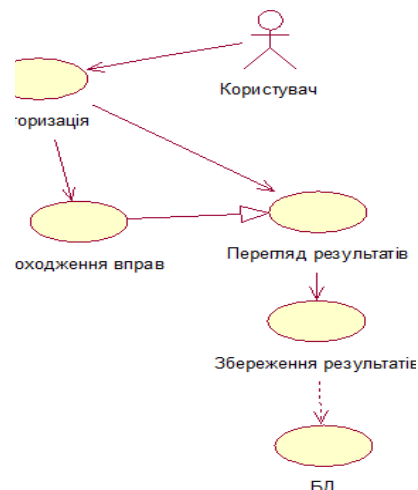


Рис. 3.2

3.5 Sequence diagram

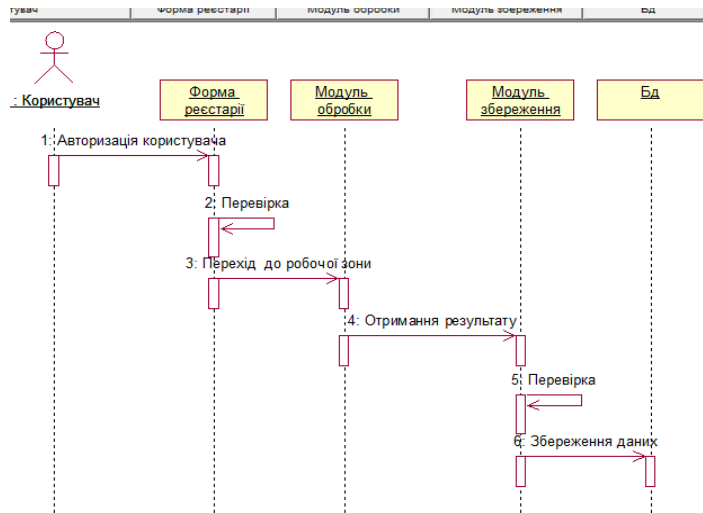


Рис. 3.3

Висновок до розділу 3

В ході виконання проектування ПП була створені блок-схеми основного та допоміжних модулів ПП, які є ядром ПП та які необхідно реалізувати на етапі кодування.

РОЗДІЛ 4 ОСНОВНІ ОСОБЛИВОСТІ РОЗРОБКИ ДОДАТКУ

4.1 Мова програмування C Sharp

C# — [об'єктно-орієнтована мова програмування](#) з безпечною [системою типізації](#) для платформи .NET. Розроблена [Андерсом Гейлсбергом](#), [Скотом Вілтамутом](#) та [Пітером Гольде](#) під егідою [Microsoft Research](#) (при фірмі [Microsoft](#)).

[Синтаксис](#) C# близький до [C++](#) і [Java](#). [Мова](#) має строгу статичну типізацію, підтримує [поліморфізм](#), перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, [винятки](#), коментарі у форматі [XML](#). Переїнявши багато що від своїх попередників — мов [C++](#), [Delphi](#), [Модула](#) і [Smalltalk](#) — C#, спираючись на практику їхнього [використання](#), виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад [множинне спадкування](#) класів (на відміну від C++)

C# є дуже близьким родичем мови [програмування Java](#). Мова Java була створена компанією Sun Microsystems, коли глобальний розвиток інтернету поставив задачу роззосереджених обчислень. Взявши за основу популярну мову C++, Java виключила з неї потенційно небезпечні речі (типу вказівників без контролю виходу за межі). Для роззосереджених обчислень була створена концепція віртуальної машини та машинно-незалежного байт-коду, свого роду посередника між вихідним текстом програм і апаратними інструкціями комп'ютера чи іншого інтелектуального пристрою.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

29

Java набула чималої популярності, і була ліцензована також і компанією [Microsoft](#). Але з плином часу Sun почала винуватити Microsoft, що та при створенні свого клону Java робить її сумісною виключно з платформою [Windows](#), чим суперечить самій концепції машинно-незалежного середовища виконання і порушує ліцензійну угоду. Microsoft відмовилася піти назустріч вимогам Sun, і тому з'ясування стосунків набуло статусу [судового](#) процесу. Суд визнав позицію Sun справедливою, і зобов'язав Microsoft відмовитися від позаліцензійного використання Java.

У цій ситуації в Microsoft вирішили, користуючись своєю вагою на ринку, створити свій власний аналог Java, мови, в якій корпорація стане повновладним господарем. Ця новостворена мова отримала назву C#. Вона успадкувала від Java концепції віртуальної машини (середовище .NET), байт-коду (MSIL) і більшої безпеки вихідного коду програм, плюс врахувала досвід використання програм на Java.

Нововведенням C# стала можливість легшої взаємодії, порівняно з мовами-попередниками, з кодом програм, написаних на інших мовах, що є важливим при створенні великих проектів. Якщо програми на різних мовах виконуються на платформі [.NET](#), .NET бере на себе клопіт щодо сумісності програм (тобто типів даних, за кінцевим рахунком).

Станом на сьогодні C# визначено флагманською мовою корпорації Microsoft, бо вона найповніше використовує нові можливості .NET. Решта мов програмування, хоч і підтримуються, але визнані такими, що мають спадкові прогалини щодо використання .NET.

Символ # у назві мови можна інтерпретувати і як дві пари плюсів ++, що натякають на новий крок у розвитку мови порівняно з C++ (подібно до кроку від C до C++), і як музичний символ [дієз](#), разом з буквою C, що становить в англійській мові назву ноти [до-дієз](#). Останнє й дало назву мові.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

30

Попри те, що символ <#> (октоторп) насправді є символом для позначення номера на більшості клавіатур і відрізняється від символу дієз <#> (Unicode U+266F), Microsoft, як автор мови, неодноразово зверталася до своїх клієнтів з проханням прийняти таку стилізацію.

C# розроблялась як мова програмування прикладного рівня для [CLR](#) і тому вона залежить, перш за все, від можливостей самої CLR. Це стосується, перш за все, системи типів C#. Присутність або відсутність тих або інших виразних особливостей мови диктується тим, чи може конкретна мовна особливість бути трансльована у відповідні конструкції CLR. Так, з розвитком CLR від версії 1.1 до 2.0 значно збагатився і сам C#; подібної взаємодії слід чекати і надалі. (Проте ця закономірність буде порушена з виходом C# 3.0, що є розширеннями мови, що не спираються на розширення платформи .NET.) CLR надає C#, як і всім іншим .NET-орієнтованим мовам, багато можливостей, яких позбавлені «класичні» мови програмування. Наприклад, збірка сміття не реалізована в самому C#, а проводиться CLR для програм, написаних на C# точно так, як і це робиться для програм на VB.NET, J# тощо.

Титульним [компілятором](#) C# є [Microsoft Visual C#](#).

Існують інші компілятори C#, часто вони включають реалізації Common Language Infrastructure і бібліотеки класів .NET:

- Проект Microsoft Rotor (який тепер зветься Shared Source Common Language Infrastructure, ліцензований тільки для навчального і дослідницького використання) забезпечує реалізації [CLR runtime](#) і компілятор C#, і підмножину бібліотек фреймворка Common Language Infrastructure, відповідно до специфікації ECMA (до C# 2.0, і з підтримкою тільки [Windows XP](#)).

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

31

- Проект SharpDevelop від компанії icsharpcode , який використовується як альтернатива Visual Studio . Забезпечує повну реалізацію Common Language Infrastructure. Остання стабільна версія IDE 4.4 (28 серпня 2013) , тестова версія 5.0 (13 лютого 2014. Зовнішній вигляд IDE дуже нагадує Microsoft Visual C#,що робить комфортним перехід від однієї середи до іншої.
- Проект Mono, початий компанією Xamarin і продовжений її покупцем і наступником Novell, забезпечує відкритий компілятор C#, повну відкрити реалізацію Common Language Infrastructure, включаючи потрібні бібліотеки фреймворка відповідно до специфікації ECMA, і близьку до повної реалізацію власницьких бібліотек класів Microsoft .NET до .NET 2.0, але не специфічних бібліотек .NET 3.0 і .NET 3.5, як для Mono 2.0.
- Проект DotGNU також надає відкритий компілятор C#, близьку до повної реалізацію Common Language Infrastructure, включаючи потрібні бібліотеки фреймворка відповідно до специфікації ECMA, і підмножину деяких залишених власницьких біліотек класів Microsoft .NET до .NET 2.0 (які не документовані або не включені у специфікації ECMA, але включені у стандартне визначення Microsoft .NET Framework).
- DotNetAnywhere Micro Framework Common Language Runtime націлений на вбудовані системи, і підтримує майже всі специфікації C# 2.0.

4.2 Мова програмування SQL

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

32

SQL (англ. *Structured query language* — мова структурованих запитів) — декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних і її модифікації, системи контролю за доступом до бази даних. Сам по собі SQL не є ні системою керування базами даних, ні окремим програмним продуктом. Не будучи мовою програмування в тому розумінні, як C або Pascal, SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати в якості інструкцій для керування даними. Стандарт SQL, крім того, вміщує функції для визначення зміни, перевірки і захисту даних.

SQL — це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також управління базами даних. Багато баз даних підтримує SQL з розширеннями до стандартної мови. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення, і вилучення даних, використовуючи систему управління і адміністративні функції. SQL також включає CLI (Call Level Interface) для доступу і управління базами даних дистанційно.

Перша версія SQL була розроблена на початку 1970-х років у IBM. Ця версія мала назву SEQUEL і була призначена для обробки й пошуку даних, що містилися в реляційній базі даних IBM, System R. Мова SQL пізніше була стандартизована Американськими Держстандартами (ANSI) в 1986. Спочатку SQL розроблялась як мова запитів і управління даними, пізніші модифікації SQL створено продавцями системи управління базами даних, які додали процедурні конструкції, control-of-flow команд і розширення мов. З випуском стандарту SQL:1999 такі розширення були формально запозичені як частина мови SQL через Persistent Stored Modules (SQL/PSM).

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

33

Критики SQL включає відсутність крос-платформенності, невідповідною обробкою відсутніх даних (дивіться `Null (SQL)`), і іноді неоднозначна граматики і семантика мови.

На початку 1970-х років в одній з дослідницьких лабораторій компанії IBM була розроблена експериментальна реляційна СУБД System R, для якої потім була створена спеціальна мова SEQUEL, що дозволяла відносно просто управляти даними в цій СУБД. Аббревіатура SEQUEL розшифровувалася як англ. *Structured English QUery Language* — «структурована англійська мова запитів». Пізніше з юридичних міркувань^[2] мова SEQUEL була перейменована в SQL. Коли в 1986 році перший стандарт мови SQL був прийнятий ANSI (American National Standards Institute), офіційною вимовою стало [ˌes kjuːˈel] — ес-к'ю-ел. Попри це, навіть англомовні фахівці по колишньому часто називають SQL сіквел, замість ес-кью-ел.

Метою розробки було створення простої непроцедурної мови, якою зміг би скористатися будь-який користувач, що навіть не має навиків програмування. Власне розробкою мови запитів займалися Дональд Чемберлін (Donald D. Chamberlin) і Рей Бойс (Ray Boyce). Пет Селінджер (Pat Selinger) займалася розробкою вартісного оптимізатора (англ. *cost-based optimizer*), Реймонд Лорі (Raymond Lorie) займався компілятором запитів.

Варто відзначити, що SEQUEL не був єдиною мовою подібного призначення. У Каліфорнійському університеті в Берклі була розроблена некомерційна СУБД Ingres (що була, між іншим, дальнім прародичем популярної зараз некомерційної СУБД PostgreSQL), яка була реляційною СУБД, але використовувала свою власну мову QUEL, яка, проте, не витримала конкуренції за кількістю СУБД, що підтримували його, з мовою SQL.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

34

Першими СУБД, що підтримують нову мову *SQL*, стали в 1979 році Oracle V2 для машин VAX від компанії Relational Software Inc. (що згодом стала компанією Oracle) і System/38 від IBM, заснована на System/R.

Перший офіційний стандарт мови SQL був прийнятий ANSI в 1986 і ISO (*Міжнародною організацією зі стандартизації*) в 1987 (так званий SQL-86) і дещо уточнений в 1989 році. Подальший розвиток мови постачальниками СУБД зажадав ухвалення в 1992 р. нового розширеного стандарту (ANSI SQL-92 або просто SQL2). Наступним стандартом став SQL:1999 (SQL3). В наш час діє стандарт, прийнятий в 2003 році (SQL:2003) з невеликими модифікаціями, внесеними пізніше.

Як і з багатьма стандартами, що мають місце в ІТ-індустрії, з мовою SQL виникла проблема, що у минулому багато виробників ПЗ з використанням SQL вирішили, що функціональність в поточній (на той момент часу) версії стандарту недостатня (що, в принципі, для ранніх версій SQL було певною мірою справедливо) і його бажано розширити. Що і призводить в цей час до того, що у різних виробників СУБД в ході різних діалекти SQL, в загальному випадку між собою несумісні.

До 1996 року питаннями відповідності комерційних реалізацій SQL стандарту займався в основному інститут NIST, який і встановлював рівень відповідності стандарту. Але пізній підрозділ, що займався СУБД, був розформований, і в цей час всі зусилля з перевірки СУБД на відповідність стандарту лягають на її виробника.

Вперше поняття «*Рівня відповідності*» було запропоноване в стандарті SQL-92. А саме, ANSI і NIST визначали чотири рівні відповідності реалізації цьому стандарту:

- *Entry* (базовий)

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

- *Transitional* (перехідний) — перевірку на відповідність цьому рівню проводив тільки інститут NIST
- *Intermediate* (проміжний)
- *Full* (повний)

Легко можна зрозуміти, що кожен подальший рівень відповідності свідомо мав на увазі відповідність попередньому рівню. Далі, згідно з цими рівнями стандартів будь-яка СУБД, яка відповідала рівню Entry, могла заявляти себе як «*SQL-92 відповідна*», хоча насправді переносимість і відповідність стандарту обмежувалося набором можливостей, що входять у цей рівень.

Положення змінилося з введенням стандарту SQL:1999. Відтепер стандарт придбав модульну структуру — основна частина стандарту була винесена в розділ «*SQL/Foundation*», всі інші були виведені в окремі модулі. Відповідно, залишився тільки один рівень сумісності — *Core*, що означало підтримку цієї основної частини. Підтримка решти можливостей залишена на розсуд виробників СУБД. Аналогічне положення мало місце і з подальшими версіями стандарту.

4.2.1. Переваги

- Незалежність від конкретної СУБД

Незважаючи на наявність діалектів і відмінностей в синтаксисі, в більшості своїй тексти SQL-запитів, що містять, DDL і DML, можуть бути досить легко перенесені з однієї СУБД в іншу. Існують системи, розробники яких спочатку закладалися на застосування щонайменше кількох СУБД (наприклад: система електронного документообігу Documentum може працювати як з Oracle, так і

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

36

з Microsoft SQL Server та IBM DB2). Природно, що при застосуванні деяких специфічних для реалізації можливостей такої переносимості добитися вже дуже важко.

- **Наявність стандартів**

Наявність стандартів і набору тестів для виявлення сумісності і відповідності конкретній реалізації SQL загальноприйнятому стандарту тільки сприяє «стабілізації» мови. Правда, варто звернути увагу, що сам по собі стандарт місцями занадто формалізований і роздутий в розмірах, наприклад, Core-частину стандарту SQL:2003 включає понад 1300 сторінок тексту.

- **Декларативність**

За допомогою SQL програміст описує тільки те, які дані потрібно витягнути або модифікувати. Те, яким чином це зробити, вирішує СУБД безпосередньо при обробці SQL-запиту. Проте не варто думати, що це повністю універсальний принцип — програміст описує набір даних для вибірки або модифікації, проте йому при цьому корисно уявляти, як СУБД розбиратиме текст його запиту. Особливо критичні такі моменти стають при роботі з великими базами даних і зі складними запитамі — чим складніше сконструйований запит, тим більше він допускає варіантів написання, різних за швидкістю виконання, але тих самих за набором даних.

4.2.2 Недоліки

- **Невідповідність реляційної моделі даних**

Творець реляційної моделі даних Едгар Кодд, Крістофер Дейт та їхні прихильники указують на те, що SQL не є істинно реляційною мовою. Зокрема вони указують на такі проблеми SQL:

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

- Рядки, що повторюються
- Невизначені значення (null)
- Явна вказівка порядку стовпчиків зліва направо
- Стовпці без імені та імена стовпчиків, що дублюються
- Відсутність підтримки властивості «=>»
- Використання покажчиків
- Висока надлишковість

У опублікованому Крістофером Дейтом і Г'ю Дарвенем Третьому Маніфесті вони висловлюють принципи СУБД наступного покоління і пропонують мову Tutorial D, яка є достовірно реляційною.

- Складність

Хоча SQL і замислювався, як засіб роботи кінцевого користувача, врешті-решт він став настільки складним, що перетворився на інструмент програміста.

- Відступи від стандартів

Незважаючи на наявність міжнародного стандарту ANSI SQL-92, багато компаній, СУБД (наприклад, Oracle, Sybase, Microsoft, MySQL), що займаються розробкою, вносять зміни до мови SQL, вживаної в розроблених ними СУБД, тим самим відступаючи від стандарту. Таким чином з'являються специфічні для кожної конкретної СУБД діалекти мови SQL.

- Складність роботи з ієрархічними структурами

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Раніше SQL не пропонував стандартного способу маніпуляції деревовидними структурами. Деякі постачальники СКБД пропонували свої рішення. Наприклад, Oracle використовує вираз CONNECT BY. В наш час як стандарт прийнята рекурсивна конструкція WITH.

4.3 Середовище розробки

В якості середовища для розробки програми була обрана Microsoft Visual Studio.

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Visual Studio включає один або декілька з наступних компонентів:

- Visual Basic .NET, а до його появи — Visual Basic
- Visual C++
- Visual C#
- Visual J#
- Visual F# (входить до складу Visual Studio 2010);

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

39

- Visual Studio Debugger

Багато варіантів постачання також включають:

- Microsoft SQL Server або
- MSDE Visual Source Safe — файл-серверна система управління версіями

У минулому, до складу Visual Studio також входили продукти:

- Visual InterDev;
- Visual J++;
- Visual J#;
- Visual FoxPro;
- Visual Source Safe – файл-серверна система управління версіями.

Visual Studio 2008

В листопаді 2007 корпорація Microsoft оголосила ^[4] про випуск нових продуктів для розробників Visual Studio 2008 (кодове ім'я Orcas) і .NET Framework 3.5.

Visual Studio 2008 сконцентрувала свою увагу на розробці застосунків для Windows Vista, Microsoft Office 2007 та веб – застосунків. Для візуальної розробки було презентовано Windows Presentation Foundation та новий HTML/CSS редактор. Маючи понад 250 нових функціональних можливостей, Visual Studio 2008 пропонує істотні поліпшення кожної з версій, включаючи Visual Studio Express та Visual Studio Team System.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

- Language Integrated Query (LINQ) заповнює прогалину між об'єктним програмуванням і даними та дозволяє розробникам зосередитися не на доступі до даних, а на роботі з ними.
- Visual Studio Team System підтримує управління збіркою програм, включаючи виконання планових збірок та збірок в результаті процесу безперервної інтеграції. Team Build забезпечує інтегровану підтримку статичного аналізу коду під час виконання збірки і проведення контрольних випробувань збірки.
- Значне спрощення розробки для Web завдяки новій техніці обміну інформацією з веб-сервером для веб-сайтів, які підтримують AJAX/JSON. Нові елементи управління ASP.NET передбачають покращене управління сторінками і шаблони, а Windows Communication Foundation передбачає вбудовану підтримку RSS і REST.
- .NET Framework 3.5 також містить декілька нових функціональних можливостей, серед яких можна назвати можливості для Web 2.0, сервіс-орієнтовану архітектуру (Service-Oriented Architecture, SOA) та програми на базі технології ПЗ + Сервіси (Software+Services). Сервіси з підтримкою послідовності операцій надають нові класи моделі програмування, які спрощують створення сервісів з підтримкою послідовності операцій за рахунок використання Windows Communication Foundation і Windows Workflow Foundation. Це дозволяє розробникам на .NET Framework створювати бізнес-логіку сервісу, використовуючи WF, та організовувати обмін повідомленнями з цим сервісом за допомогою WCF.
- Підтримка додаткових протоколів веб-сервісів у Windows Communication Foundation, включаючи протоколи Web Services Atomic

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

41

Transaction (WS-AtomicTransaction) 1.1, WS-ReliableMessaging 1.1, WS-Secure Conversation та Web Services Coordination (WS-Coordination) 1.1.

- Text Template Transformation Toolkit шаблонно орієнтований генератор коду включений як частина середовища.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

42

РОЗДІЛ 5 КЕРІВНИЦТВО КОРИСТУВАЧА

5.1 Як встановити ПП

Щоб встановити даний ПП, необхідно розмістити файл в робочу папку та встановити програму SQL Server Configuration Manager.

5.2 Як зробити резервні копії БД

Файл-> Резервне копіювання БД.

5.3. Інструкція для користувача

Перед відкриттям додатку, користувачу потрібно запустити сервер.

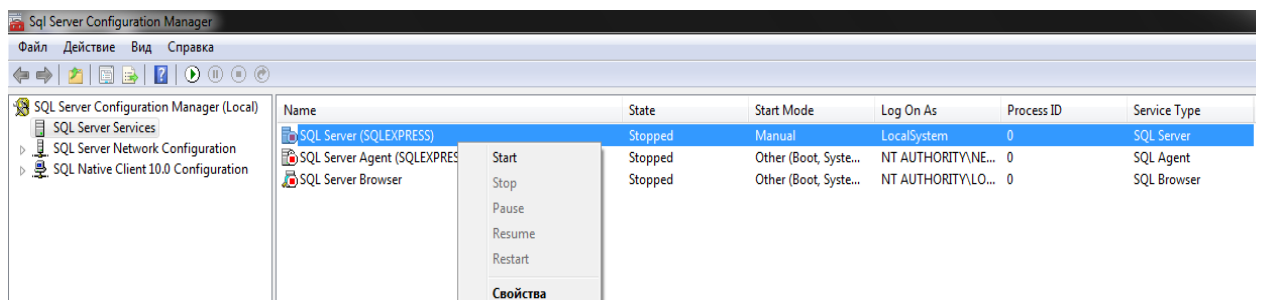


Рисунок 5.1 SQL Server

При запуску ПП відкривається вікно авторизації, де необхідно ввести логін та пароль для входу в систему, або ж зареєструватися, як новий користувач, що зображено на рисунку 5.2 – 5.3.

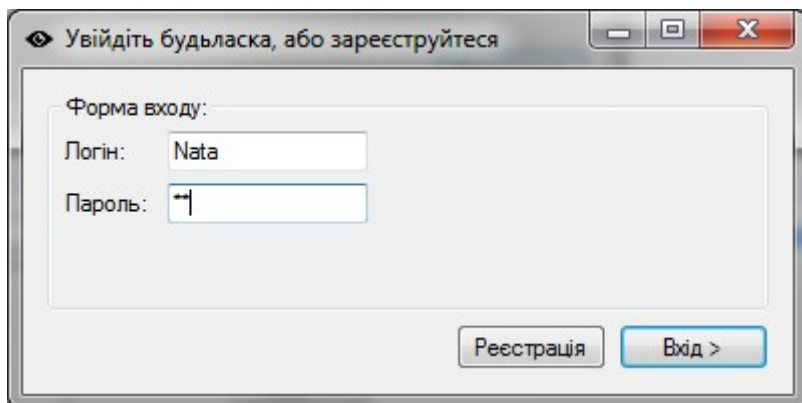


Рисунок 5.2 – Вікно авторизації

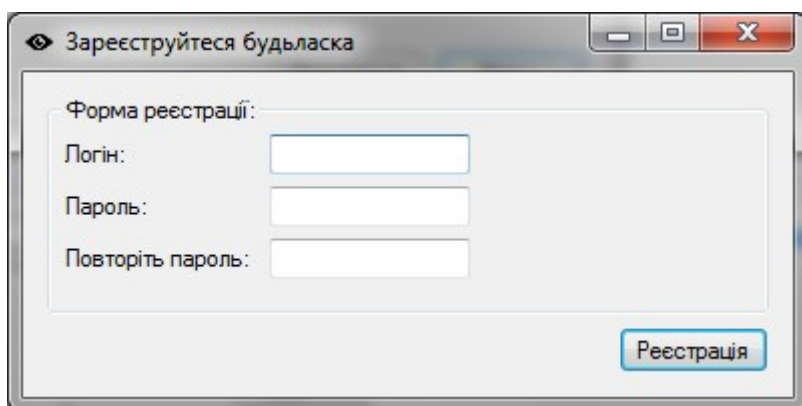


Рисунок 5.3 – Вікно реєстрації

Після реєстрації, користувач заходить в головне меню додатку

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

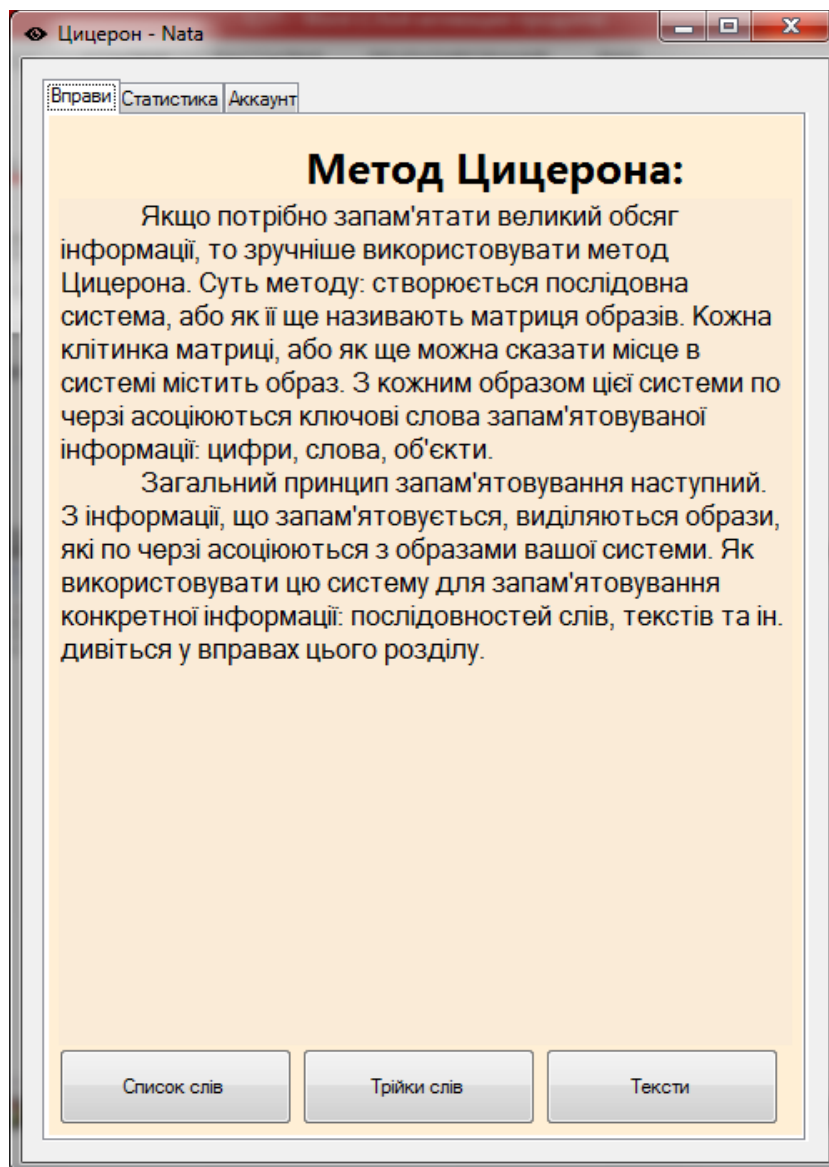


Рисунок 5.4 Головне вікно.

«Список слів», «Трійки слів» та «Тексти» - це вправи, які користувач може проходити. Перейшовши до будь якого з цих вікон, він спочатку читає про вправу, як її проходити, потім вибирає кількість слів та час, за який він запам'ятає послідовно слова та потім напише їх.

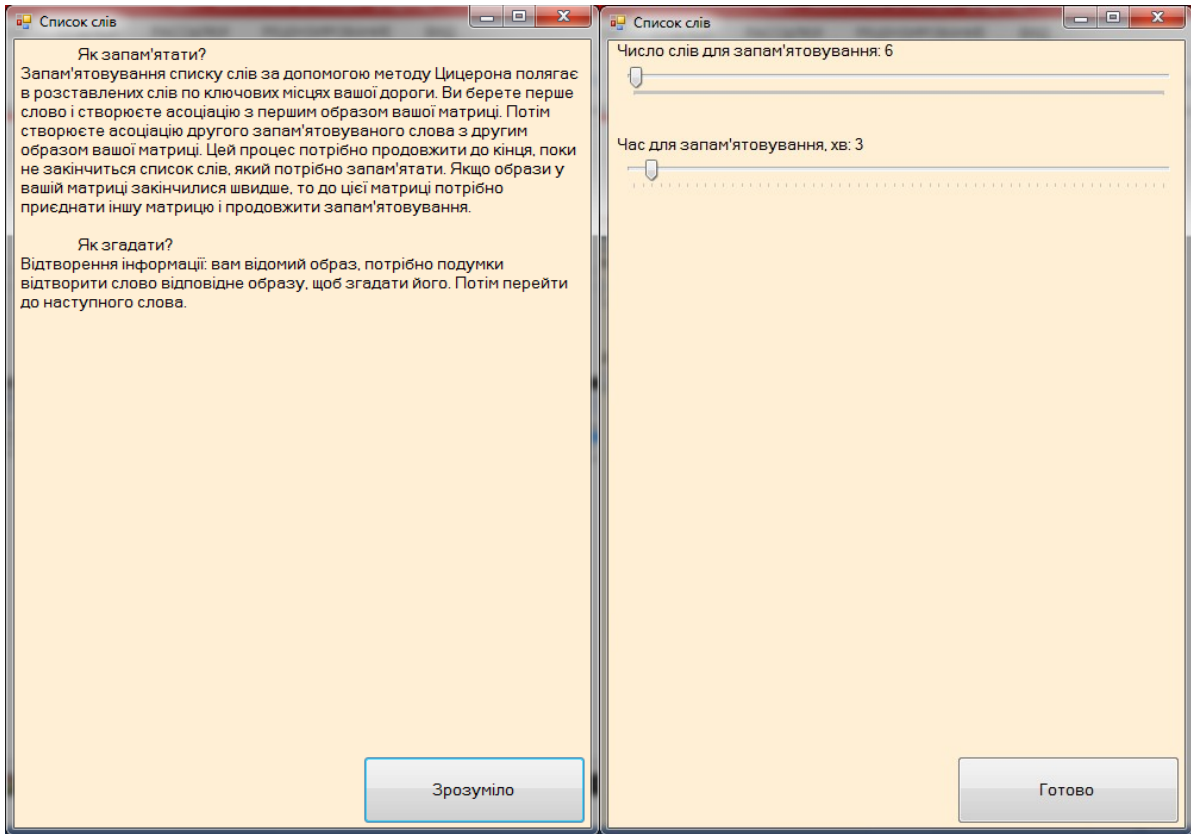


Рисунок 5.5 «Список слів»

Рисунок 5.6 «Список слів»

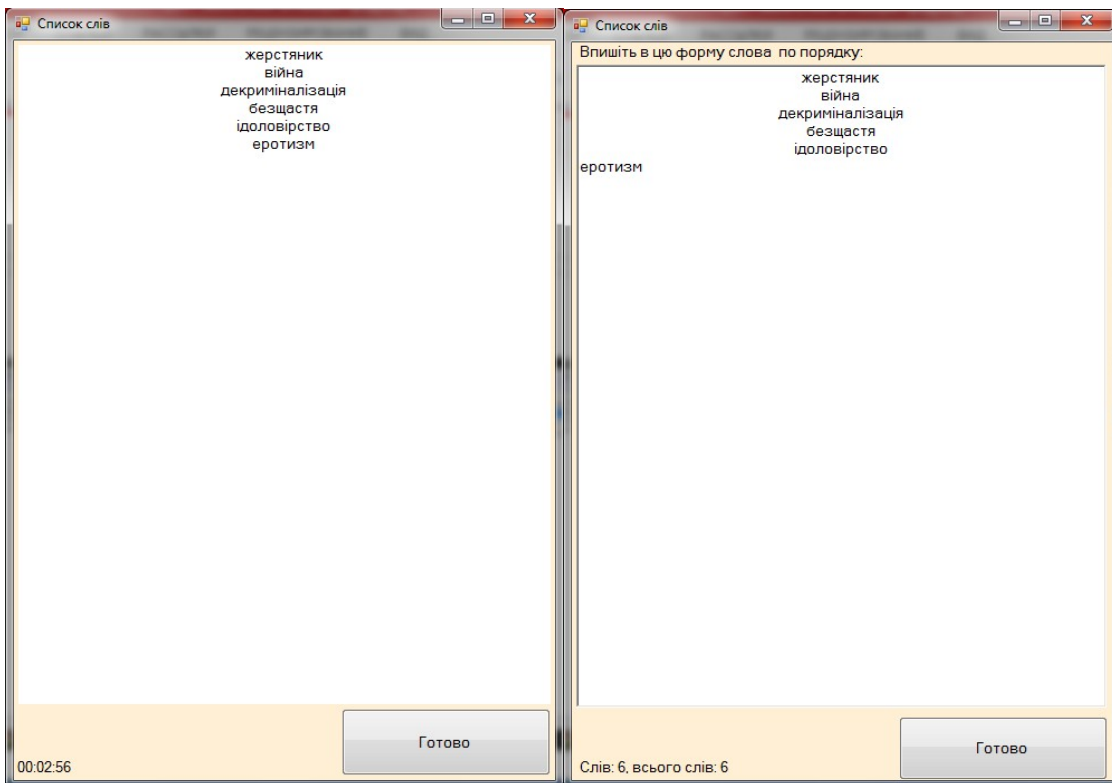


Рисунок 5.7 «Список слів» - слова
для запам'ятовування

Рисунок 5.8 «Список слів» - вікно
для запису слів

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

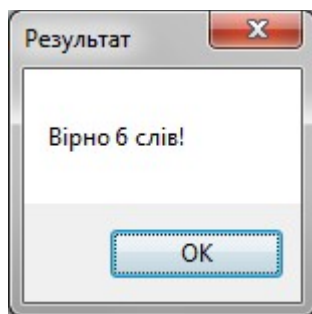


Рисунок 5.9 Результат

В полі «Статистика» відображаються результати проходження вправ користувача. Користувач може вибрати тип діаграм та вибірку даних, по яких побудувати графік.

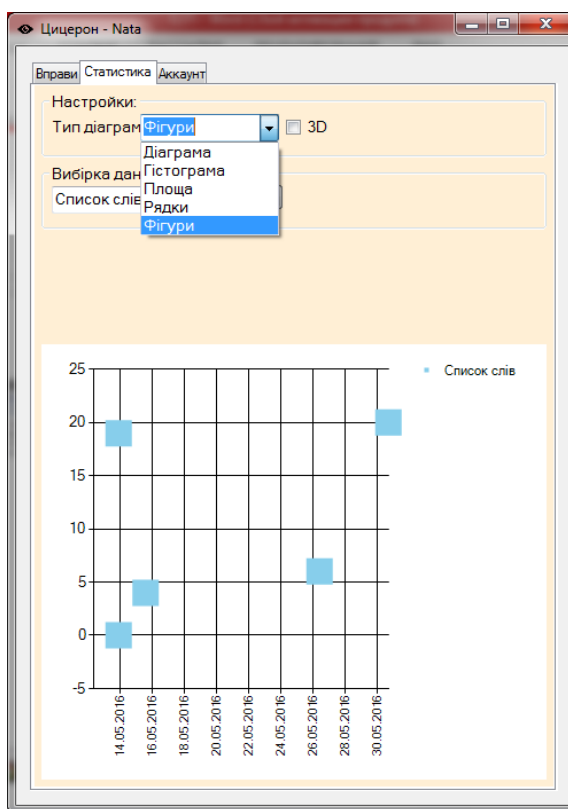


Рисунок 5.10 «Статистика»

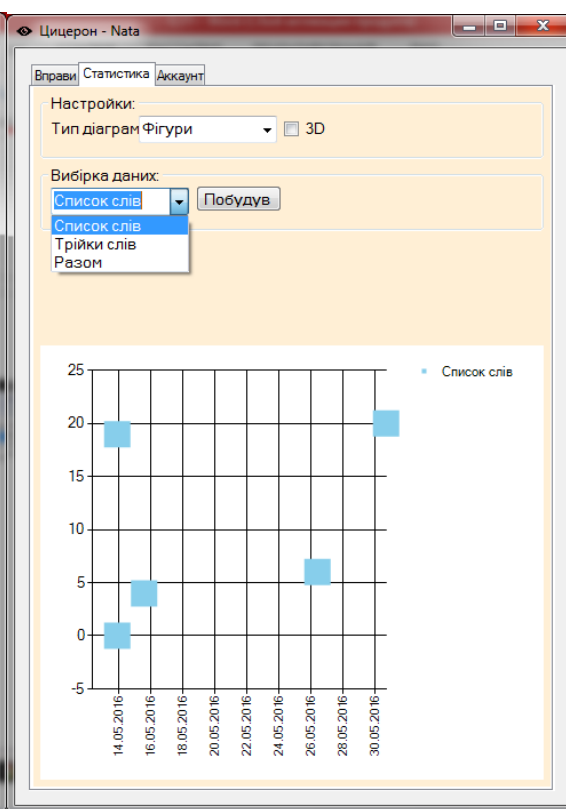


Рисунок 5.11 «Статистика»

В полі «Аккаунт» користувач може зміни пароль, чи логін.

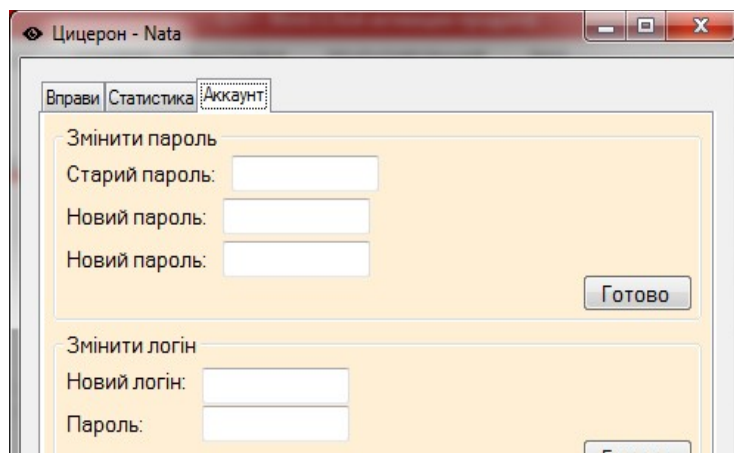


Рисунок 5.12 «Аккаунт»

Після закриття додатку, користувач має вимкнути сервер.

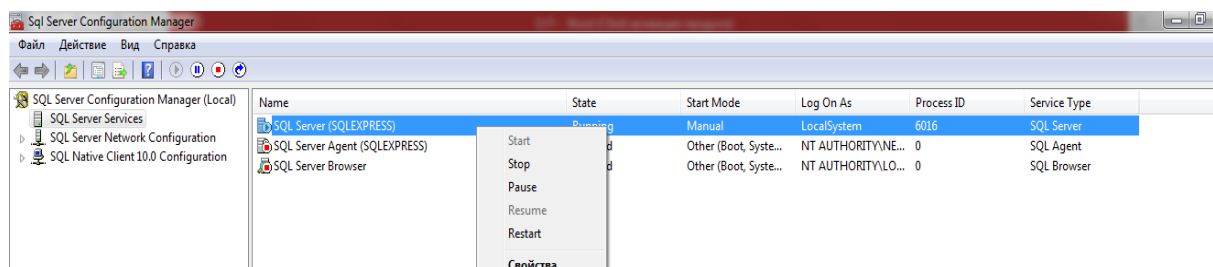


Рисунок 5.13 SQL Server

Висновки до розділу 5

В розробленому керівництві користувача для даного ПП розглянуто всі етапи від установки та початку роботи до завершення та всі режими роботи. Наведені правила користування ПП.

РОЗДІЛ 6 ОХОРОНА ПРАЦІ

Вступ

Додаток для оцінки та розвитку пам'яті по методу Цицерона призначений для моніторингу та покращення пам'яті користувача. В даному розділі проводиться аналіз умов праці у кабінеті що займається обслуговуванням користувачів та фіксуванням даних, використовуючи ПЗ.

6.1 Аналіз умов на робочому місці

6.1.1 Характеристика приміщення

Розглядаються умови праці в реальному приміщенні, яке є офісним і знаходиться на другому поверсі триповерхової будівлі, має розмір. В приміщенні є одне вікно розмірами 2 , один кондиціонер.

Схема розміщення робочих місць показана на рисунку 6.1

Таблиця 6.1 – Характеристики приміщення

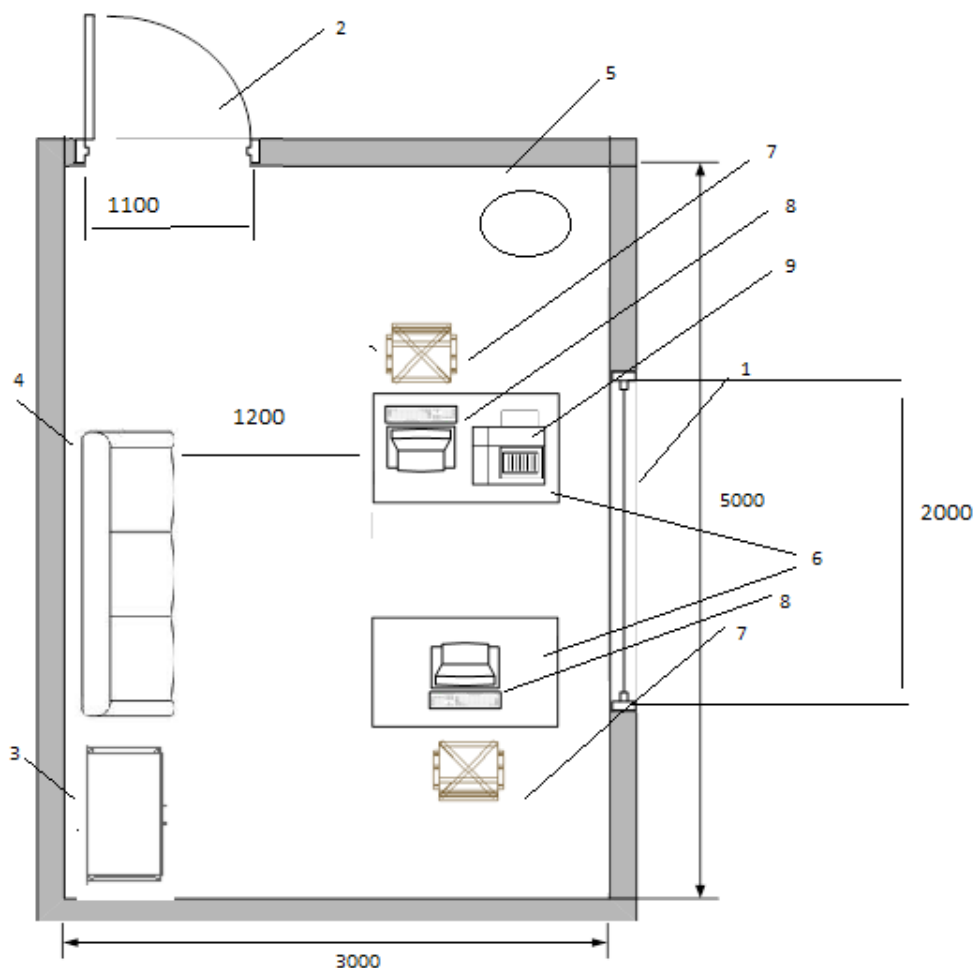
Параметри	Кількісна і візуальна характеристика
Розміри приміщення	3000 (мм) × 5000 (мм) × 3000 (мм)
Кількість працюючих	2 чол.
Площа	3,0 (м) × 5,0 (м) = 15,0 (м ²)
Об'єм кабінета	3,0 (м) × 5,0 (м) × 3,0 (м) = 45,0 (м ³)
Диван	60x200x50

Таблиця 6.2 – Характеристики ПК Impression Ultra A0311

Короткі технічні характеристики	AMD Llano A6-3650 (2.06 (ГГц))/ RAM 2 (ГБ) / HDD 500 (ГБ) / ATI Radeon HD6530 / DVD+/-RW
Процесор	Чотирьох'ядерний AMD Llano A6-3650 (2.6 (ГГц))
Об'єм оперативної пам'яті	2 (ГБ)
Тип пам'яті	DDR3-1333
Тип відеокарти та об'єм відеопам'яті	Інтегрована, ATI Radeon HD6530
Об'єм HDD	500 (ГБ)
Оптичний привід	DVD-RW

Таблиця 6.3 – Характеристики рідкокристалічного монітору Philips E-line 227E3LSU/00 LED

Розмір дисплею	21.5 "
Ключові технології	LED-підсвітка
Відношення сторін	16:9
Яскравість дисплею	250 (кд/м ²)
Розмір пікселя	0.248 (мм)
Максимальне розширення дисплею	1920 x 1080
Споживана потужність	В активному режимі: 18,5 (Вт) В режимі очікування: 0,5 (Вт)
Габарити монітору, маса	Розміри: З підставкою: 534 x 408 x 216 (мм) Без підставки: 534 x 336 x 41 (мм) Маса: З підставкою: 3,68 (кг) Без підставки: 3,46 (кг)



1 – вікно, 2 – двері, 3 – шафа, 4 – диван, 5 – вазон, 6 – стіл, 7 – стул, 8 – радіокристалічний монітор, 9 – принтер

Рисунок 6.1 – План приміщення

При розміщенні обладнання в приміщенні слід дотримуватись норм ДСанПІН 3.3.2.007-98 та Положення про умови зберігання архівних документів. У даному приміщенні норми розміщення та відстаней наведені у табл. 6.4 та табл. 6.5.

Таблиця 6.4 – Норми відстаней за ДСанПІН 3.3.2.007-98

Параметр	Нормативне значення	Реальне значення
Прохід	Не менше 1000 (мм)	1200(м)
Двері	1000 (мм) × 2100 (мм)	1100 (мм) × 2100 (мм)
Вікно	1200 (мм) × 2000 (мм)	1600 (мм) × 2000 (мм)
Площа на одну людину	Не менше 6 (м ²)	7,5 (м ²)

Зм	Лист	№ докум.	Підп.	Дата
----	------	----------	-------	------

ЛД21.9102.1300.1732.ПЗ

Аркуш

51

Об'єм на одну людину	Не менше 20 (м ³)	22,5 (м ³)

6.2 Оцінка небезпечних та шкідливих виробничих факторів

У цьому пункті розглядаються заходи для покращення та забезпечення нормалізації умов праці при роботі з розробленим програмним забезпеченням. Небезпечні та шкідливі виробничі фактори відповідно до ГОСТ 12.0.003-74 за природою дії поділяються на 4 групи (табл. 8.7).

Таблиця 6.7 – Небезпечні та шкідливі захисні фактори

Фізичні	Біологічні	Психофізіологічні
Мікроклімат, освітлення, шум, випромінення, електронебезпека, пожежонебезпека	Віруси, бактерії	відсутні

Біологічні фактори

Види та засоби захисту від небезпечних та шкідливих хімічних факторів наведені у табл. 6.9.

Таблиця 6.9 – Небезпечні та шкідливі біологічні фактори

Вид захисту		Засоби
Технічні заходи	У технологічному обладнанні	Корпус персонального комп'ютера, вільна циркуляція повітря
	У приміщенні	Підтримується в зразковому порядку та чистоті, що виключало б можливість появи плісняви,

		комах та гризунів.
Організаційні заходи	<p>З метою своєчасного виявлення плісневих грибів і комах необхідно двічі на рік (початок і кінець опалювального сезону) проводити вибіркоче мікробіологічне та ентомологічне обстеження документів та приміщень сховищ. Забороняється перебування у верхньому одязі, брудному взутті, зберігання будь-яких сторонніх предметів, зберігання і вживання харчових продуктів, а також розведення квітів.</p>	

6.3 Мікроклімат

Таблиця 6.10 – Основні характеристики мікроклімату приміщення

Показники	Реальні значення	Нормативні значення
Відносна вологість повітря	52 %	40...60 %
Температура повітря:		
– в холодний період року	23 °С	21...25 °С
– в теплий період року	26 °С	22...28 °С
Швидкість переміщення повітря	0,08 м/с	0,1 м/с
Інтенсивність теплового випромінювання	29 Вт/м ²	35 Вт/м ²

За санітарно-гігієнічним нормуванням (ДСН 3.3.6.042-99) основні характеристики мікроклімату аудиторії відповідають встановленим нормам, виконано наступні заходи і засоби для нормалізації параметрів мікроклімату (табл. 6.11).

Таблиця 6.11 – Заходи для нормалізації параметрів мікроклімату

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Вид захисту		Засоби
Технічні заходи	У технологічному обладнанні	Застосовуються куллери у персональному комп'ютері
	У приміщенні	У холодну пору року: Опалення здійснюється за допомогою одного радіатора М-140. Кількість секцій – 10. У теплу пору року: Кондиціонер типу спліт-системи, марки Delfa DSR-07HR (режим охолодження/обігрів/осушення; потужність, яка споживається – 0,73 (кВт), потужність в режимі охолодження – 2,1 (кВт)).
Організаційні заходи		Вологе прибирання, кондиціонування, зволоження або осушення повітря

6.4 Освітлення

Таблиця 6.12 – Характеристика зорової роботи

Характеристика зорової роботи	Найменший розмір об'єкту розрізнення, (мм)	Розряд зорової роботи	Підрозряд зорової роботи	Контраст об'єкта розрізнення з фоном	Характеристика фону	Шрифти	Природне освітлення
							бокове
Середньої точності	Понад 0,5 до 1	IV	в	Малий Середній Великий	Світлий Світлий Середній	1 (мм) 2-3 (мм) 4-5 (мм)	1,5

Зм	Лист	№ докум.	Підп.	Дата
----	------	----------	-------	------

ЛД21.9102.1300.1732.ПЗ

Аркуш

54

Природне освітлення – бокове (однобічне), що здійснюється через світловий отвір (вікно) у зовнішній стіні.

Природне освітлення забезпечується одним вікном, що виходить на схід, розміром 1,6 (м) × 2,0 (м) та з білими тканинними ролетами. Склопакет у металопластикових вікнах двокамерний.

Штучне освітлення у даному приміщенні є комбінованим – здійснюється за допомогою двох світильників у верхній зоні приміщення та світильниками, що концентрують світловий потік безпосередньо на робочих місцях.

Штучне освітлення забезпечується за допомогою світильників з люмінесцентними лампами серії ЛПО-01, типу люмінесцентних ламп білого світла ЛБ 40, зі світло проникаючими боковинами, форма розсіювача – прямокутна, кількість ламп – 4, потужністю 40 (Вт), які розміщені на висоті 2,9 (м) від підлоги, що відповідає вимогам нормативних документів (ДБН В.2.5-28-2006 та ДСанПіН 3.3.2.007-98 або ДНАОП 0.00-1.31-99 - не менше 2,5 (м)).

6.5 Шум

У приміщенні джерелами шуму є системний блок ПК, багатофункціональний пристрій, кондиціонер, зовнішній шум. Дані наведені у табл. 5.13.

Таблиця 6.13 – Джерела шуму

Джерела шуму	Рівень шуму	Нормативне значення
Системний блок ПК	50 дБА	50 дБА
Багатофункціональний пристрій	52 дБА	
Кондиціонер	42 дБА	
Телефон	80 дБА	

Зовнішній шум	40 дБА	
---------------	--------	--

Середній рівень шуму у приміщенні перевищує встановлені норми за ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку».

Для запобігання впливу шуму вжито наступні заходи (табл. 6.14).

Таблиця 6.14 – Заходи і засоби захисту від шуму

Вид захисту		Засоби
Технічні заходи	У технологічному обладнанні	Куллери знаходяться у системному блоці комп'ютера, відсутні зазори у з'єднаннях
	У приміщенні	Для захисту від зовнішнього шуму встановлена шумоізоляція, вікно виходить на вулицю з обмеженням швидкості, розташовується на відстані більш ніж 30 м від проїзної частини
Організаційні заходи		Режим праці і відпочинку, дотримуються правила технічної експлуатації, проведення планово-попереджувальних оглядів та ремонтів
ЗІЗ		Не передбачені

6.6 Пожежна безпека

Таблиця 6.15 – Характеристика пожежонебезпечної зони

Тип пожежі	Характеристика
Клас пожежі	А – горіння твердих речовин, Е - горіння електро-установок, під напругою до 1000 В
Підклас пожежі	А ₁ горіння супроводжується тлінням (А ₂ обернено до А ₁)

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

56

Вибухопожежонебезпечна	Категорія В (пожежонебезпечна)
Пожежонебезпечна зона	Клас - II-IIa
Горючі матеріали	Волокнисті (папір), тверді (столи, стільці, двері, шафи), пластикові (вікно, стільці, лінолеум, комп'ютер)

Для запобігання пожежі у приміщенні слід вжити наступні заходи протипожежної безпеки зазначені в табл. 6.19.

Таблиця 6.16 – Заходи протипожежної безпеки

Вид захисту	Засоби подолання небезпеки
Технічні заходи	У приміщенні знаходиться 2 вогнегасники ОУ-5. Вільний доступ до мережних рубильників та вимикачів. На стелі встановлено датчик теплової пожежної сигналізації КИ-1. У коридорі, навпроти реєстратури, знаходиться пожежний кран та рукав.
Організаційні заходи	Інструктаж з пожежної безпеки та періодичний контроль знань про правила пожежної безпеки. План евакуації при пожежі. Вільний доступ до вимикача електроживлення.
ЗІЗ	Не передбачені

Відповідно до будівельних норм та правил СНІП 2.09.02-85, шляхи евакуації людей при пожежі для даного приміщення відповідають встановленим нормам.

Висновки до розділу 6

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

57

У даному розділі дипломної роботи була розглянуто приміщення з усім оснащенням, розглянуті норми та заходи з охорони праці й техніки безпеки. На підставі опрацьованої літератури з охорони праці, були зазначені норми для забезпечення оптимальних умов мікроклімату, освітлення, рівня шуму, випромінювання, умов для дотримання правил біологічної, хімічної, пожежної безпеки та електробезпеки.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

58

ЗАГАЛЬНІ ВИСНОВКИ

В результаті виконання роботи розроблено програмний продукт у середовищі розробки Visual Studio мовою С#. Програмний продукт створений для тренування пам'яті, а також моніторингу вихідних параметрів користувача з можливістю подальшого збереження та візуалізації даних.

Запропонований програмний продукт має у своєму складі високу надійність та зручний інтерфейс користувача.

Таким чином, в ході роботи було виконано наступні задачі:

- а) розроблено простий та доступний інтерфейс додатку;
- б) надано можливість використовувати дані системи для аналізу покращення запам'ятовування.

Пояснювальна записка до дипломної роботи оформлена відповідно до Положення про державну атестацію студентів НТУУ «КПІ».

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

59

ПЕРЕЛІК ПОСИЛАНЬ

1. Лурія А.Р. Маленька книжка про велику пам'яті. Вид. МДУ, 2008.
2. Іванов-Муромський К.А. Мозок і пам'ять. Київ, Наука, 2007.
3. Артінян Г.А, Пирогов А.А. Механізми пам'яті ЦНС. Л., Наука, 2008
4. <http://o.zao.com.ua/u-chomu-poljagae-metod-cicerona/>
5. Джон Скит. С# для професіоналов: тонкості програмування, 3-е издание, новий переклад = С# in Depth, 3rd ed.. — М.: «Вільямс», 2014. — 608 с. — ISBN 978-5-8459-1909-0.
6. Кристиан Нейгел і др. С# 5.0 і платформа .NET 4.5 для професіоналов = Professional С# 5.0 and .NET 4.5. — М.: «Диалектика», 2013. — 1440 с. — ISBN 978-5-8459-1850-5.
7. А. Хейлсберг, М. Торгерсен, С. Вилтамут, П. Голд. Язык програмування С#. Класика Computers Science. 4-е издание = С# Programming Language (Covering С# 4.0), 4th Ed. — СПб.: «Питер», 2012. — 784 с. — ISBN 978-5-459-00283-6.
8. Э. Стилмен, Дж. Грин. Изучаем С#. 2-е издание = Head First С#, 2ed. — СПб.: «Питер», 2012. — 704 с. — ISBN 978-5-4461-0105-4.
9. М. Эллис, Б. Строуструп. Справочное руководство по языку С++ с комментариями: Пер. с англ. - Москва: Мир, 1992. 445с.
10. Стенли Б. Липпман. С++ для начинающих: Пер. с англ. 2тт. - Москва: Унитех; Рязань: Гэлион, 1992, 304-345сс.
11. Бруно Бабэ. Просто и ясно о Borland С++: Пер. с англ. - Москва: БИНОМ, 1994. 400с.

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

60

12. В.В. Подбельский. Язык C++: Учебное пособие. - Москва: Финансы и статистика, 1995. 560с.
13. Ирэ Пол. Объектно-ориентированное программирование с использованием C++: Пер. с англ. - Киев: НИИПФ ДиаСофт Лтд, 1995. 480с.
14. Т. Фейсон. Объектно-ориентированное программирование на Borland C++ 4.5: Пер. с англ. - Киев: Диалектика, 1996. 544с.
15. Т. Сван. Освоение Borland C++ 4.5: Пер. с англ. - Киев: Диалектика, 1996. 544с.
16. Г. Шилдт. Самоучитель C++: Пер. с англ. - Санкт-Петербург: BHV-Санкт-Петербург, 1998. 620с.
17. У. Сэвитч. C++ в примерах: Пер. с англ. - Москва: ЭКОМ, 1997. 736с.
18. К. Джамса. Учимся программировать на языке C++: Пер. с англ. - Москва: Мир, 1997. 320с.
19. В.А. Скляр. Язык C++ и объектно-ориентированное программирование: Справочное издание. - Минск: Вышэйшая школа, 1997. 480с.
20. Х. Дейтел, П. Дейтел. Как программировать на C++: Пер. с англ. - Москва: ЗАО "Издательство БИНОМ", 1998. 1024с.
21. Ицик Бен-Ган - Microsoft SQL Server 2008. Основы T-SQL
22. Ржеуцкая С.Ю. - Базы данных. Язык SQL
23. К. Дж. Дейт - SQL и реляционная теория. Как грамотно писать код на SQL
24. Бейли Л. - Изучаем SQL
25. Гурвиц Г.А. - Microsoft Access 2010. Разработка приложений на реальном примере

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

61

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

62

ДОДАТОК А

Фрагмент коду ГОЛОВНОГО ВІКНА

```
using System;
using System.Windows.Forms;

namespace NatashaDiplom
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private void MainForm_Load(object sender, EventArgs e)
        {
            Visible = false;
            DialogResult login = new LoginForm().ShowDialog();
            switch (login)
            {
                case DialogResult.Cancel:
                    this.Close();
                    break;
                case DialogResult.OK:
                    this.Visible = true;
                    this.Text = "Цицерон - " + Models.User.Name;
                    //////////////////////////////////////
                    break;
            }
        }

        private void buttonTrinity_Click(object sender, EventArgs e)
        {
            Visible = false;
            DialogResult dr = new FormTrinity().ShowDialog();
            switch (dr)
            {
                default:
                    this.Visible = true;
                    break;
            }
        }

        private void buttonList_Click(object sender, EventArgs e)
```

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ

Аркуш

63

```
{
    Visible = false;
    DialogResult dr = new FormList().ShowDialog();
    switch (dr)
    {
        default:
            this.Visible = true;
            break;
    }
}

private void buttonText_Click(object sender, EventArgs e)
{
    Visible = false;
    DialogResult dr = new FormText().ShowDialog();
    switch (dr)
    {
        default:
            this.Visible = true;
            break;
    }
}
}
```

Зм	Лист	№ докум.	Підп.	Дата

ЛД21.9102.1300.1732.ПЗ