

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-06

Дипломний проект

здобувача освіти денної форми навчання

РП.06.21.000.ДП

**ТОМАШЕВСЬКИЙ АРТЕМ
ВІТАЛІЙОВИЧ**

**м. Одеса
2023 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-06

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка програмного забезпечення системи планування домашніх справ
"Сімейний органайзер"

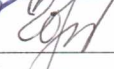
Проектний матеріал складається з пояснювальної записки на 69 сторінках та графічного (презентаційного) матеріалу на 14 аркушах (слайдах).

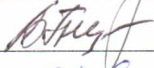
Дипломник  (Томашевський А.В.)

Керівник  (Селіванова А. В.)

Консультанти:

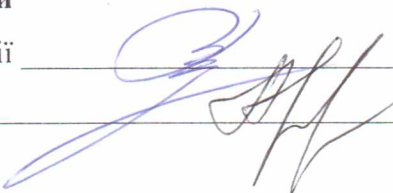
з економічної частини  (Копайгородська Т. Г.)

з охорони праці  (Чорновол Н. І.)

з дотримання вимог ЄСКД  (Петрашова В. І.)

старший консультант  (Кунуп Т. В.)

До захисту допущений

Голова циклової комісії  (Кривченко Ю. В.)

Завідувач відділення  (Скорнякова О. В.)

Захист « 23 » 06 2023 р. Протокол ДКК № 2

Оцінка ДКК 3 (задовільно)

Секретар ДКК 

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Ігор БЕРКАНЬ

“ ” 2023 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Томошевському Артему Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка програмного забезпечення системи планування домашніх справ "Сімейний органайзер"

затверджена наказом по коледжу від “ 17 ” жовтня 2022 р. № 235-А2-ОД

2. Термін здачі закінченого проекту (роботи) 9 червня 2023

3. Вихідні дані до проекту (роботи) Технічне завдання, тестові питання для формування завдань, Visual Studio, C#

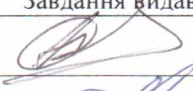
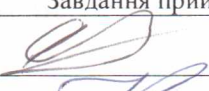
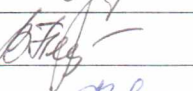
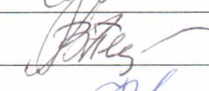
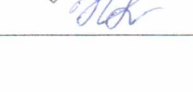
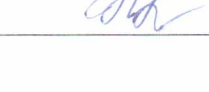
4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Огляд предметної галузі, аналіз технологій; Мета розробки, постановка задачі, призначення та основні параметри, вибір функціональних елементів реалізації системи; вибір технологій та інструментів реалізації системи; основні етапи проектування; опис етапів реалізації програмного продукту; тестування

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

Мультимедійна презентація 11 слайдів: тема, виконавець; актуальність; мета, задачі; аналіз аналогів; засоби реалізації; особливості

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Технологічний	Селіванова А. В.		
Економічна частина	Копайгородська Т. Г.		
Охорона праці	Чорновол Н. І.		
Нормоконтроль	Петрашова В. І.		
Старший консультант	Кунуп Т. В.		

7. Дата видачі завдання

15.05.2023

Керівник

Завдання прийняв до виконання

(підпис)

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Аналітичний огляд предметної галузі, обґрунтування актуальності теми	16.05.2023	Виконано
2	Аналіз технологій та аналогічних систем	16.05.2023	Виконано
3	Визначення мети, постановка задачі,	16.05.2023	Виконано
4	Вибір технологій та інструментів реалізації системи	21.05.2023	Виконано
5	Моделювання програмного забезпечення	31.05.2023	Виконано
6	Проектування програмного забезпечення	31.05.2023	Виконано
7	Реалізація програмного забезпечення	31.05.2023	Виконано
8	Доопрацювання та отримання основних результатів проектування	31.05.2023	Виконано
9	Робота над розділом «Економічна частина»	05.06.2023	Виконано
10	Робота над розділом «Охорона праці»	05.06.2023	Виконано
11	Робота над графічною частиною дипломного проекту. Створення мультимедійної презентації проекту.	08.06.2023	Виконано

Дипломник

(підпис)

Керівник

(підпис)

[illegible]

ЗМІСТ:

ПЕРЕДМОВА.....	7
1. ТЕХНОЛОГІЧНИЙ РОЗДІЛ.....	9
1.1 Існуючі застосунки для планування особистого часу.....	9
1.2 Технічне завдання.....	10
1.3 Етапи розробки програмного забезпечення "Сімейний органайзер"	11
1.4 Обґрунтування вибору середовища та мови програмування.....	13
1.4.1 Мова програмування C#.....	13
1.4.2 Windows Forms.....	14
1.4.3 Робота з файлами в C#.....	15
1.5 Архітектура програмного забезпечення.....	18
1.5.1 Клас Form1.....	18
1.5.2 Обробники подій.....	21
1.5.3 Взаємодія з елементами керування.....	22
1.6 Розробка функціоналу програмного забезпечення "Сімейний органайзер"	
1.6.1 Створення нотаток.....	23
1.6.2 Відображення нотаток за датою.....	25
1.6.3 Збереження нотаток.....	28
1.7 Створення інтерфейсу користувача.....	33
1.7.1 Головне вікно додатку.....	33
1.7.2 Елементи керування форми.....	35
1.8. Тестування та валідація додатку "Сімейний органайзер".....	38
1.8.1 План тестування.....	38
1.8.2 Результати тестування.....	39
2. ЕКОНОМІЧНА ЧАСТИНА.....	43
3. ОХОРОНА ПРАЦІ.....	49
Висновки	54
Список використаних джерел.....	55

ПЕРЕДМОВА

Органайзер являється засобом керування часом. Попереднє планування справ допомагає підвищити ефективність будь-якої діяльності, як особистої, так і професійної. Комп'ютерна програма-органайзер — це комп'ютерна програма, що відноситься до прикладного програмного забезпечення і призначена для накопичення інформації користувача, а потім оперативного пошуку по ній, організації справ і контролю за їх виконанням, відслідковування визначених користувачем подій. Являється однією із форм персонального органайзера. Написання даних програм можливе різноманітними мовами програмування і в різних середовищах розробки. Функції типової комп'ютерної програми-органайзера пов'язані із забезпеченням роботи наступних підрозділів: календар; записна книжка; менеджер контактів(адресно-телефонна книга); події, прив'язані до певної дати(свята, зустрічі); годинник; калькулятор. Сфери застосування і вимоги до органайзерів досить різноманітні, тому існують різні версії, які можуть не містити деяких функцій, вказаних вище, проте мати певні специфічні додатки, необхідні конкретній категорії користувачів. Особисті інформаційні системи (програма-секретар). Вони дозволяють проводити планування особистого часу, своєчасно нагадують про початок запланованих заходів, ведуть персональні та інші карточки з можливістю автоматичної вибірки інформації, ведуть персональні інформаційні записники для збереження різноманітної особистої інформації.

Органайзер – програма, призначена для накопичення інформації користувача, а потім оперативного пошуку по ній, організації справ і контролю за їх виконанням, відслідковування визначених користувачем подій. Завданням роботи є створення програми – органайзера для планування робочого часу, ведення записної книжки, управління пріоритетами нотаток, нагадування. До складу додатка повинні входити: записна книжка, годинник, календар.

					<i>РП 06. 21 000. ДП ПЗ</i>	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		7

Захистити додаток від копіювання шляхом прив'язки до параметра в системному реєстрі. Створити консольний додаток, оптимізований для розповсюдження на різних версіях операційних систем Windows. Основними операціями, виконуваними програмою, є: робота з нотатками, встановлення їх пріоритетності та часове нагадування.

Додаток надасть можливість швидко зберігати та знаходити необхідну інформацію, управляти пріоритетами та нагадуваннями, що позитивно позначиться на повсякденному житті.

У розділах проекту будуть розглянуті вимоги, проектування, реалізація, тестування та можливі напрямки розвитку додатку. Використовуватимуться мова програмування C# та технологія Windows Forms для створення графічного інтерфейсу.

					<i>РП 06. 21 000. ДП ПЗ</i>	Арк.
						8
Зм.	Арк	№ докум.	Підпис	Дата		

1. ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1.1 Існуючі застосунки для планування особистого часу

Any.do забезпечує можливість до кожного завдання додавати теги, нагадування, підзавдання та вкладення. Функція пріоритетів допоможе сконцентруватися на головному та не забути про важливу справу, а списками можна ділитися з друзями та колегами – та планувати спільні проєкти. У платній версії можна створювати завдання, що повторюються, прикріплювати файли будь-якого розміру, використовувати індивідуальні теми, а також створювати нагадування, прив'язані до певного місця.

Todoist – головний конкурент та відмінна альтернатива Any.do. У додатку немає хіба що дерева, що самознищується. У Todoist можна розбивати завдання щодо проєктів та підпроєктів, додавати теги, змінювати тему, додавати коментарі, прикріплювати файли та вести спільні проєкти. Як додаткова мотивація в сервіс вбудований трекінг продуктивності. Деякі функції, включаючи нагадування та додавання завдань по e-mail, доступні лише у преміум-версії програми.

Wunderlist - ще один застосунок для планування особистого часу. Незважаючи на те, що Microsoft, який купив Wunderlist, заявив про закриття програми, сервіс досі працює. Сам Wunderlist залишається досить зручною програмою: до кожного завдання можна додати нагадування, повтори, підзадачі, нотатки та файли, створювати кілька різних списків та налаштовувати їх під себе. Wunderlist може похвалитися ще однією особливістю - функцією «Бесіди», завдяки якій можна обговорювати спільні завдання, не виходячи із програми.

Викупивши популярний Wunderlist, Microsoft не стали винаходити колесо і створили додаток Microsoft To-Do, що практично повністю повторює можливості Wunderlist. У Microsoft To-Do додали кілька нових функцій: індивідуальний планувальник завдань, списки з позначеннями кольорів і пропозиції, які дозволяють ефективніше планувати час.

Google Календар - один з найпростіших і найнадійніших планувальників, що синхронізується з усіма розглянутими програмами. Якщо раніше в календарі можна було лише розсортувати завдання по днях та часу, то зараз додалося функцій. Наприклад, у Google Календарі з'явилася можливість ставити цілі: задаєте ціль, і програма сама пропонує вам час для її досягнення.

GTasks - ще один дуже простий сервіс для шанувальників мінімалізму. Програма дозволяє створювати завдання, призначати їм час, ділитися ними з колегами та друзями. У безкоштовному додатку є реклама, яка може неабияк дратувати. Повний аналог програми без реклами – TickTick.

Trello - дуже зручна для роботи над проєктами в невеликих командах, непогано підходить і для особистого планування справ. У Trello можна створити віртуальну дошку з картками, до карток додати учасників, мітки, чеклист, термін виконання, вкладення та змінити обкладинку.

MyLifeOrganized - програма з легким інтерфейсом, яка підтримує всі функції якісного планувальника завдань: оформлення списків, додавання підзадач, нотаток, тегів та нагадувань. На жаль, створити відразу кілька списків або обговорити завдання з колегами всередині MyLifeOrganized не вдасться. Якщо у вас багато завдань, краще вибрати інше рішення [1].

Аналітичний огляд сімейних органайзерів є важливою частиною проектування та розробки семейного органайзера, оскільки дозволяє оцінити існуючі рішення та визначити їх переваги та недоліки. Цей аналіз надає цінну інформацію для визначення позиціонування розроблюваного органайзера на ринку та визначення його конкурентних переваг.

Основна мета аналітичного огляду полягає в ідентифікації сильних та слабких сторін існуючих рішень.

Мобільний додаток "Family Organizer Plus"

Переваги:

1. Інтуїтивний інтерфейс та зручне користування.
2. Функціонал для спільного планування подій та завдань.

					РП 06. 21 000. ДП ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		10

3. Можливість створення спільних списків покупок та контролю над ними. Легкість використання та інтуїтивний інтерфейс.

4. Функціонал, спеціально розроблений для сімейного планування, включаючи календар, списки завдань, сповіщення та спільні нотатки.

5. Можливість синхронізації даних з різними пристроями та відстеження активностей різних членів сім'ї.

Недоліки:

1 Вимагає наявності мобільного пристрою та доступу до Інтернету.

2 Відсутність можливості перегляду та редагування на більшому екрані, наприклад, на комп'ютері.

3 Можливість обмеженого доступу до деяких функцій в безкоштовній версії додатку.

4 Відсутність можливості синхронізації з іншими календарями.

5 Обмежена функціональність для контролю над фінансами сім'ї.

6 Відсутність можливості створювати власні шаблони для подій та завдань.

Мобільний додаток "Family Organizer" є зручним інструментом для сімейного планування. Він пропонує простий та інтуїтивний інтерфейс, що полегшує користування. Функціонал додатка спеціально орієнтований на потреби сімей, включаючи календар, списки завдань, сповіщення та спільні нотатки. Завдяки можливості синхронізації даних, кожен член сім'ї може бачити та відстежувати активності інших членів.

Проте, для використання додатку необхідний мобільний пристрій та доступ до Інтернету, що може бути обмеженням для деяких користувачів. Крім того, відсутність можливості перегляду та редагування на більшому екрані може бути незручною, особливо для складних завдань або детального планування. Деякі функції можуть бути доступні лише у платній версії додатку, що може обмежувати безкоштовних користувачів у використанні повного функціоналу.

Онлайн-сервіс "FamilyHub"

Переваги:

Широкий функціонал для організації сімейного життя, включаючи календар, списки завдань, спільні замітки та файли.

Можливість додавати фотографії та спільно редагувати контент.

Синхронізація з іншими пристроями та календарями.

Широкий функціонал для організації сімейного життя, включаючи календар, списки завдань, спільні замітки та файли.

Можливість додавати фотографії та спільно редагувати контент.

Синхронізація з іншими пристроями та календарями.

Недоліки:

3 Складний процес налаштування та використання.

4 Платний доступ до деяких функцій та обмеження по кількості користувачів.

5 Відсутність можливості персоналізації інтерфейсу.

6 Складний процес налаштування та використання.

7 Платний доступ до деяких функцій та обмеження по кількості користувачів.

8 Відсутність можливості персоналізації інтерфейсу.

Онлайн-сервіс "FamilyHub" має значний функціонал, який сприяє організації сімейного життя. Він надає можливість створювати та спільно керувати календарем, списками завдань, спільними замітками та файли. Також можна додавати фотографії та спільно редагувати контент. Синхронізація з іншими пристроями та календарями сприяє зручному використанню сервісу.

Проте, сервіс "FamilyHub" має свої недоліки. Перш за все, процес налаштування та використання може бути складним, особливо для менш технічно освічених користувачів. Деякі функції та можливості можуть бути доступні лише за певну плату або мають обмеження по кількості користувачів. Крім того, сервіс може не надавати широкі можливості

персоналізації інтерфейсу, що обмежує користувача у налаштуванні зовнішнього вигляду та розташування елементів.

Розроблений сімейний органайзер враховує ці недоліки і пропонує простий інтерфейс, широкий функціонал, можливість синхронізації та персоналізації, що робить його бажаною альтернативою для користувачів, які шукають зручний та ефективний інструмент для сімейного планування.

Друкований сімейний планувальник

Переваги:

1. Фізичний формат, що дозволяє візуально організувати розклад та плани сім'ї.
2. Можливість швидко додавати записи та замітки.
3. Відсутність потреби у доступі до Інтернету або мобільних пристроїв.
4. Фізичний формат, що дозволяє візуально організувати розклад та плани сім'ї.
5. Можливість швидко додавати записи та замітки.
6. Відсутність потреби у доступі до Інтернету або мобільних пристроїв.

Недоліки:

1. Відсутність автоматичної синхронізації з іншими пристроями.
2. Важкість вносити зміни та виправлення.
3. Обмежені можливості спільної роботи та доступу для кількох користувачів.
4. Відсутність автоматичної синхронізації з іншими пристроями.
5. Важкість вносити зміни та виправлення.
6. Обмежені можливості спільної роботи та доступу для кількох користувачів.

Розглянутий друкований сімейний планувальник має свої переваги, такі як фізичний формат, який дозволяє зручно організовувати розклад та плани сім'ї. Він також не вимагає доступу до Інтернету або мобільних

пристроїв, що може бути зручним у деяких ситуаціях. Однак, він має обмежені можливості спільної роботи та синхронізації з іншими пристроями, що ускладнює спільне планування для всієї сім'ї. Також, внесення змін та виправлень у фізичному форматі може бути складним та часомістким. З цими обмеженнями, розроблений семейний органайзер забезпечує більш гнучкі та зручні можливості організації та спільної роботи для сімейного планування.

1.2 Технічне завдання

Постановка завдання є першим кроком у розробці проекту "Сімейний органайзер". У цьому розділі буде розглянуто постановка завдання та описано основні функціональні вимоги до програмного забезпечення.

Для початку проаналізуємо потреби користувачів, що будуть вирішені за допомогою цього додатку. Вона полягає у необхідності створення та збереження нотаток, які можуть містити текстову інформацію для подальшого використання. Ця програма дозволить користувачеві зручно керувати своїми нотатками та легко знайти їх за датою.

На основі цих потреб були визначені основні *функції* до програмного забезпечення. Основними функціями додатку є:

Можливість створювати нові нотатки та вводити в них текстову інформацію.

Можливість зберігати нотатки для подальшого використання.

Можливість відображення нотаток, відсортованих за датою створення.

Інтерфейс користувача, що дозволяє легко навігувати та взаємодіяти з додатком.

Окрім цього, враховуються і *додаткові функції*, такі як:

Надійність: додаток повинен зберігати нотатки без втрати даних.

Простота використання: інтерфейс користувача повинен бути зрозумілим і легким для навігації.

Швидкодія: додаток повинен працювати ефективно, забезпечуючи швидке створення, збереження та відображення нотаток.

					РП 06. 21 000. ДП ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		14

Функціональні вимоги

Створення нотаток: Користувач повинен мати можливість створювати нові нотатки, вводити текстовий контент та зберігати його для подальшого використання.

Відображення нотаток за датою: Додаток повинен забезпечувати відображення нотаток, відсортованих за датою їх створення. Користувач повинен мати можливість переглядати нотатки, відносячись до конкретної дати.

Збереження нотаток: Користувач повинен мати можливість зберігати створені нотатки для збереження їх на постійне зберігання та доступу в майбутньому.

Нефункціональні вимоги

Надійність: Додаток повинен бути надійним та стабільним у роботі, не викликати аварійне завершення та відновлюватись після помилок.

Швидкодія: Додаток повинен працювати ефективно та реагувати на користувацькі дії швидко, без помітних затримок.

Сумісність: Додаток повинен бути сумісним з різними операційними системами, забезпечуючи його коректну роботу на різних платформах.

Безпека: Додаток повинен забезпечувати захист конфіденційної інформації, запобігаючи несанкціонованому доступу до нотаток.

1.3 Етапи розробки програмного забезпечення "Сімейний органайзер"

Мета дослідження цього проекту "Сімейний органайзер" полягає у розробці функціонального програмного забезпечення, яке надасть користувачам зручний і ефективний спосіб створення, збереження та управління своїми нотатками. Головною метою є створення інтуїтивного і легкого у використанні інструменту, що дозволить користувачам організувати свої ідеї, завдання та іншу важливу інформацію у вигляді нотаток.

					РП 06. 21 000. ДП ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		15

Для досягнення цієї мети проект передбачає виконання наступних завдань:

1. Розробка інтерфейсу користувача: Основний фокус завдання буде приділено розробці зручного і привабливого графічного інтерфейсу, який буде легко навчати та використовувати користувачам будь-якого рівня досвіду. Вивчення ергономіки і засад проектування інтерфейсів допоможе забезпечити зручність та ефективність взаємодії користувача з програмою.

2. Реалізація функціональності створення нотаток: Основна функція додатку полягатиме у створенні нотаток користувачем. Завдання буде спрямовано на розробку механізму для створення, редагування та форматування текстових нотаток, а також додаткових функцій, які можуть полегшити користування, наприклад, можливість додавання зображень або відео до нотаток.

3. Реалізація функціональності збереження нотаток: Важливим аспектом проекту буде розробка механізму збереження нотаток. Завдання буде спрямовано на вивчення методів збереження даних, включаючи використання файлової системи для збереження нотаток у вигляді текстових файлів.

4. Реалізація функціональності управління нотатками: Проект також передбачає розробку механізму управління нотатками, зокрема, можливості перегляду, редагування та видалення наявних нотаток. Завдання буде спрямовано на розробку зручних інструментів для пошуку та сортування нотаток, а також організацію нотаток за категоріями або мітками.

5. Забезпечення надійності і безпеки даних: Важливим аспектом дослідження буде забезпечення надійності і безпеки даних користувачів. Вивчення методів резервного копіювання, обробки помилок та захисту від несанкціонованого доступу допоможе забезпечити захист важливої інформації користувачів.

6. Тестування та валідація: Проект включає етап тестування, який допоможе перевірити правильність реалізації функціональності та

ідентифікувати помилки або недоліки. Валідація програмного забезпечення здійснюватиметься шляхом проведення користувацьких опитувань та отримання зворотного зв'язку від користувачів.

1.4 Обґрунтування вибору середовища та мови програмування

1.4.1 Мова програмування C#

Мова програмування C# є однією з основних мов програмування, використовуваних для розробки програмного забезпечення на платформі .NET. Вона була розроблена компанією Microsoft і базується на мові програмування C. C# поєднує в собі простоту використання, потужність і гнучкість, що робить її популярним вибором для розробників.

Основні особливості мови C# включають:

1. **Об'єктно-орієнтоване програмування:** C# підтримує концепцію об'єктно-орієнтованого програмування, що дозволяє створювати класи, об'єкти, успадкування, поліморфізм та інші конструкції, які сприяють модульності та повторному використанню коду.

2. **Сильна типізація:** Мова C# має сильну типізацію, що означає, що типи даних повинні бути визначені перед використанням, а приведення типів між різними типами має бути явно вказане.

3. **Збирання сміття:** C# використовує механізм автоматичного збирання сміття, що дозволяє автоматично видалити об'єкти, які більше не використовуються, звільняючи пам'ять і полегшуючи роботу розробника.

4. **Розширені можливості бібліотеки класів .NET:** C# надає доступ до багатьох багатофункціональних бібліотек класів .NET, які містять готові рішення для роботи зі стрічками, масивами, колекціями, введенням-виведенням, мережами, базами даних та іншими ресурсами.

5. **Підтримка асинхронного програмування:** Сучасні версії мови C# надають потужні засоби для асинхронного програмування з використанням ключових слів `async` і `await`. Це дозволяє ефективно працювати з

багатопоточними операціями, забезпечуючи швидке та ефективне виконання програми.

Використання мови C# у проєкті дозволяє реалізувати надійні, ефективні та масштабовані програмні рішення. Її потужність і гнучкість дозволяють легко виконувати різноманітні завдання і задовольняти потреби користувачів.

1.4.2 Windows Forms

Windows Forms є одним із ключових компонентів розробки графічного інтерфейсу користувача в середовищі розробки програмного забезпечення для платформи .NET. У цьому розділі ми розглянемо основні аспекти Windows Forms та його роль у реалізації нашого проєкту "Сімейний органайзер".

Windows Forms надає розробникам можливість створювати вікна, форми та елементи керування (такі як кнопки, тексти, полоси прокрутки тощо), щоб створити інтерактивний інтерфейс користувача для своїх програм. Одна з основних переваг Windows Forms полягає в його простоті використання та широкому спектрі доступних функціональних можливостей.

Створення вікон та форм у Windows Forms відбувається шляхом використання класу Form. Цей клас є базовим для всіх форм додатку і містить властивості та методи для налаштування зовнішнього вигляду та поведінки форми. Наприклад, властивість Text дозволяє встановлювати заголовок форми, а властивість Size визначає її розміри.

Створення елементів керування відбувається за допомогою різних класів, які відповідають кожному типу елемента керування. Наприклад, клас Button представляє кнопку, клас TextBox - текстове поле, а клас ListBox - список для вибору елементів. Ці класи надають розробникам доступ до властивостей та методів для налаштування поведінки та вигляду кожного елемента.

Один з основних принципів Windows Forms - подійно-орієнтоване програмування. Кожен елемент керування може генерувати різні події, такі як натискання кнопки, введення тексту або вибір елемента списку. Розробник може підписатися на ці події та виконати певний код у відповідь на них. Наприклад, можна виконати дії після натискання кнопки або зміни значення текстового поля.

Windows Forms також надає можливості для роботи з файлами та потоками даних. У нашому проекті ми використовуємо класи StreamReader і StreamWriter з простору імен System.IO для зчитування та записування текстових даних в файл.

Клас StreamReader дозволяє нам відкривати файл для читання та зчитувати його з використанням різних методів, наприклад, ReadLine, який зчитує наступний рядок тексту з файлу. ми використовуємо цей клас для зчитування попередніх записів нотаток з файлу, щоб відобразити їх у текстовому полі.

Клас StreamWriter дозволяє нам відкривати файл для запису та записувати дані в нього, такі як нові нотатки, які вводить користувач. ми використовуємо цей клас для створення або оновлення файлу з нотатками, зберігаючи нові записи в кінці файлу.

Окрім цього, Windows Forms надає можливості для роботи з подіями форми. Наприклад, ми використовуємо подію Form.Load, яка виникає при завантаженні форми, для виконання певних налаштувань та ініціалізації даних перед відображенням форми.

Основні аспекти Windows Forms та їх роль у реалізації проекту "Сімейний органайзер" полягає у створенні форм та елементів керування, роботи з файлами та подіями форми. Знання цих концепцій допоможе нам глибше розуміти процес створення інтерфейсу користувача та манпулювання даними у нашому проекті.

1.4.3 Робота з файлами в С#

Робота з файлами в С# є важливою частиною розробки програмного забезпечення, в тому числі і для нашого сімейного органайзера. Для забезпечення збереження та завантаження нотаток і подій ми використовуємо різні методи та класи з .NET Framework.

1. Створення файлу:

```
string filePath = "notes.txt"; // шлях до файлу
// Створення нового файлу з використанням класу File та методу Create
if (!File.Exists(filePath))
{
    File.Create(filePath).Close();
}
```

2. Запис у файл:

```
string textToWrite = "Привіт, це моя перша нотатка!";
// Запис тексту у файл з використанням класу StreamWriter
using (StreamWriter writer = new StreamWriter(filePath))
{
    writer.WriteLine(textToWrite);
}
```

3. Читання з файлу:

```
// Читання тексту з файлу з використанням класу StreamReader
using (StreamReader reader = new StreamReader(filePath))
{
    string textRead = reader.ReadToEnd();
    Console.WriteLine(textRead);
}
```

4. Додавання даних до вже існуючого файлу:

```
string newText = "Це нова нотатка для додавання.";
// Додавання тексту до файлу з використанням класу StreamWriter з
параметром true
```

```
using (StreamWriter writer = new StreamWriter(filePath, true))
{
    writer.WriteLine(newText);
}
```

5. Видалення файлу:

```
// Видалення файлу з використанням методу Delete класу File
if (File.Exists(filePath))
{
    File.Delete(filePath);
}
```

Ці приклади показують базові операції з файлами в C#. Залежно від потреб проекту, можуть бути використані інші класи та методи для роботи з файлами, такі як BinaryReader/BinaryWriter для бінарних файлів або XmlReader/XmlWriter для роботи з XML.

1.5 Архітектура програмного забезпечення

1.5.1 Клас Form1

Клас Form1 є головним класом нашої програми, який відповідає за візуальний інтерфейс користувача. Він включає в себе різноманітні компоненти, такі як кнопки, тексти, календарі та списки, що дозволяють користувачу взаємодіяти з програмою.

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {

```

					<i>РП 06. 21 000. ДП ПЗ</i>	Арк.
						21
Зм.	Арк	№ докум.	Підпис	Дата		


```

// Ініціалізація компонентів та налаштування інтерфейсу
InitializeComponents();

// Завантаження нотаток та подій з файлу
LoadNotesFromFile();
LoadEventsFromFile();

// Відображення нотаток та подій
DisplayNotes();
DisplayEvents();
}

private void btnAddNote_Click(object sender, EventArgs e)
{
    // Обробка натискання кнопки "Додати нотатку"
    AddNote();
}

private void btnAddEvent_Click(object sender, EventArgs e)
{
    // Обробка натискання кнопки "Додати подію"
    AddEvent();
}

private void calendar_SelectionChanged(object sender, EventArgs e)
{
    // Обробка зміни вибраної дати в календарі
    UpdateSelectedDate();
}

// Додаткові методи та функції для роботи з нотатками та подіями

private void InitializeComponents()
{
    // Ініціалізація компонентів форми
    // Налаштування розміщення та розмірів елементів інтерфейсу

```

```

        // Додавання подій обробки до необхідних компонентів
    }

    private void LoadNotesFromFile()
    {
        // Завантаження нотаток з файлу та збереження їх відповідних
        об'єктів
    }

    private void LoadEventsFromFile()
    {
        // Завантаження подій з файлу та збереження їх відповідних
        об'єктів
    }

    private void DisplayNotes()
    {
        // Відображення списку нотаток відповідно до обраної дати
    }

    private void DisplayEvents()
    {
        // Відображення списку подій відповідно до обраної дати
    }

    private void AddNote()
    {
        // Додавання нової нотатки та оновлення списку нотаток
    }

    private void AddEvent()
    {
        // Додавання нової події та оновлення списку подій
    }

```

```

private void UpdateSelectedDate()
{
    // Оновлення обраної дати та відображення нотаток та подій для
    нової дати
}

```

Клас `Form1` містить різні методи, які відповідають за обробку подій, завантаження даних з файлу, відображення і оновлення списків нотаток та подій, а також додавання нових нотаток та подій. Кожен метод виконує певну функцію, що сприяє зручному управлінню подіями та нотатками в сім'ї.

1.5.2 Обробники подій

Обробка подій є важливим аспектом нашого сімейного органайзера, оскільки вона дозволяє реагувати на дії користувача та виконувати відповідні дії. Використовуючи мову програмування C# та технологію Windows Forms, ми реалізуємо обробники подій для різних елементів інтерфейсу.

Наприклад, для кнопок "Додати подію" або "Додати нотатку" ми додаємо обробник подій "Click", який буде виконувати відповідні дії при натисканні на ці кнопки. Нижче наведений приклад коду для обробки події натискання кнопки "Додати подію":

```

private void addButton_Click(object sender, EventArgs e)
{
    // Виконати дії при натисканні кнопки "Додати подію"
    // Наприклад, відкрити форму для додавання нової події
}

```

Також, ми можемо додати обробники подій для елементів, які відповідають за вибір дати та часу події. Наприклад, для елемента DateTimePicker ми можемо додати обробник події "ValueChanged", який буде

виконуватись при зміні вибраної дати або часу. Нижче наведений приклад коду для обробки події зміни значення DateTimePicker:

```
private void dateTimePicker_ValueChanged(object sender, EventArgs e)
{
    // Виконати дії при зміні вибраної дати або часу
    // Наприклад, оновити список подій або нотаток залежно від нового
    значення DateTimePicker
}
```

Обробники подій дозволяють нам реагувати на дії користувача та виконувати відповідні дії в нашому сімейному органайзері, забезпечуючи більшу функціональність та взаємодію з користувачем.

1.5.3 Взаємодія з елементами керування

У цьому розділі описуються методи та код, які відповідають за взаємодію з елементами керування на формі. Наприклад, зчитування тексту з елемента richTextBox1, запис до файлу, очищення вмісту richTextBox1 та інші дії, які пов'язані з маніпуляціями з елементами керування.

Робота з файлами

Взаємодія з елементами керування в сімейному органайзері реалізована за допомогою обробки подій. Кожен елемент інтерфейсу має призначені обробники подій, які виконуються при взаємодії користувача з цим елементом.

Наприклад, для кнопок "Додати подію" та "Додати нотатку" ми можемо використовувати подію "Click" для виклику функцій, які додають нові події або нотатки до органайзера.

Ось приклад коду, який показує взаємодію з кнопкою "Додати подію":

```
// Оголошуємо обробник події
private void AddEventButton_Click(object sender, EventArgs e)
{
    // Виклик функції для додавання події
}
```

```

        AddEvent();
    }
    // Функція для додавання нової події
    private void AddEvent()
    {
        // Код для додавання нової події до органайзера
        // ...
    }
    // Прив'язка обробника події до кнопки
    AddEventButton.Click += AddEventButton_Click;

```

Аналогічним чином, ми можемо взаємодіяти з іншими елементами, такими як списки, текстові поля, календарі і т.д., за допомогою відповідних подій та обробників.

Взаємодія з елементами керування в програмі дозволяє користувачам зручно взаємодіяти з органайзером та виконувати потрібні дії шляхом обробки їх дій.

1.6 Розробка функціоналу програмного забезпечення "Сімейний органайзер"

1.6.1 Створення нотаток

У цьому розділі буде розглянуто функціональність створення нотаток у додатку "Сімейний органайзер". Створення нотаток є однією з основних операцій, які користувач може виконати за допомогою додатку. Нотатки представляють собою текстові записи, які користувач може ввести в спеціальне поле.

Створення нотаток передбачає такі кроки:

Користувач відкриває додаток і бачить головне вікно, яке містить елементи керування.

Користувач знаходить поле для введення тексту, яке призначено для створення нової нотатки.

Користувач вводить текст нотатки в поле для введення. Він може використовувати клавіатуру для набору тексту.

Після введення тексту користувач має можливість зберегти нотатку, натиснувши відповідну кнопку або виконавши певну дію, що ініціює збереження.

При збереженні нотатки її текст записується в файл з відповідною назвою і розширенням ".txt". Файл зберігається на локальному пристрої користувача.

Створені нотатки будуть доступні для перегляду та редагування в подальшому. Користувач може створити будь-яку кількість нотаток, і вони будуть зберігатись відокремлено одна від одної.

Продовжуючи тему створення нотаток, розглянемо додаткові функціональні можливості, пов'язані з цим процесом.

Користувач може захотіти створити нотатку з пустим текстом або текстом, який містить лише пробіли. Додаток повинен забезпечити перевірку таких ситуацій і попереджати користувача про неприпустимість створення порожніх нотаток.

Крім простого тексту, користувач може використовувати додаткові можливості форматування, такі як жирний шрифт, курсив, підкреслення або розмітка списком. Додаток має надати відповідний інтерфейс для виконання цих дій.

Користувач може також вставляти зображення або файли в нотатку. Додаток має підтримувати цю можливість і забезпечувати зручний спосіб вставки зображень або файлів.

Після створення нотатки користувач може зберегти її, щоб забезпечити збереження змінений або новий текст. Збереження нотатки оновлює файл, що відповідає цій нотатці, з новим текстом.

Це був детальний опис процесу створення нотаток у додатку "Сімейний органайзер". Цей процес є ключовим для роботи з додатком і дозволяє користувачам створювати і зберігати свої власні записи.

Для забезпечення функціоналу створення нотаток у нашому сімейному органайзері, ми використовуємо наступні елементи коду:

```
// Оголошення змінної для збереження назви нотатки
string noteTitle;

// Оголошення змінної для збереження тексту нотатки
string noteText;

// Обробник події натискання кнопки "Створити нотатку"
private void CreateNoteButton_Click(object sender, EventArgs e)
{
    // Отримання введеної назви нотатки з текстового поля
    noteTitle = noteTitleTextBox.Text;

    // Отримання введеного тексту нотатки з текстового поля
    noteText = noteTextTextBox.Text;

    // Створення нової нотатки і додавання її до списку нотаток
    Note newNote = new Note(noteTitle, noteText);
    noteList.Add(newNote);

    // Очищення полів введення для назви та тексту нотатки
    noteTitleTextBox.Clear();
    noteTextTextBox.Clear();

    // Оновлення списку нотаток на екрані
    UpdateNoteList();
}
```

Вищенаведений код демонструє обробку події натискання кнопки "Створити нотатку". При натисканні кнопки, отримуються дані з текстових полів для назви та тексту нотатки, створюється новий об'єкт нотатки, який додається до списку нотаток. Після цього поля введення очищаються, а список нотаток оновлюється на екрані.

Таким чином, користувачі можуть зручно створювати нові нотатки у нашому сімейному органайзері та зберігати необхідну інформацію для майбутнього використання.

1.6.2 Відображення нотаток за датою

В даному пункті описано функціональність програми, яка дозволяє відображати нотатки відповідно до вибраної користувачем дати. Цей функціонал важливий для швидкого доступу до нотатків, створених в певні дні або періоди часу.

Опис реалізації:

Коли користувач змінює вибрану дату на елементі керування "monthCalendar1", спрацьовує обробник події "monthCalendar1_DateChanged".

У цьому обробнику події відбуваються наступні кроки:

1. Значення вибраної дати перетворюється у довге форматоване значення типу рядка за допомогою методу "e.Start.ToLongDateString()".
2. Очищається вміст елемента керування "richTextBox1", щоб підготувати його для нового відображення нотаток.
3. Формується ім'я файлу шляхом додавання розширення ".txt" до значення змінної "patch".
4. Використовуючи блок "using", створюється екземпляр "StreamWriter", який буде використовуватися для запису в файл. Якщо файл існує, використовується "File.AppendText(file)" для додавання до файлу, в іншому випадку використовується "File.CreateText(file)" для створення нового файлу.
5. Вміст елемента керування "richTextBox1" записується в файл за допомогою методу "fileStream.WriteLine(richTextBox1.Text)".
6. Створюється екземпляр "StreamReader" для читання вмісту файлу.
7. Вміст файлу читається по рядках, і кожний рядок додається до елемента керування "richTextBox1" за допомогою методу "richTextBox1.AppendText(line)".
8. Потік читання файлу закривається.
9. Після виконання цих кроків, елемент керування "richTextBox1" міститиме нотатки, які були створені відповідно до вибраної дати.
10. Користувач зможе переглянути ці нотатки та редагувати їх у відповідному елементі керування.

11. При наступних змінах вибраної дати, процес буде повторено, і елемент керування "richTextBox1" буде оновлюватись з відповідними нотатками для нової дати.

Цей функціонал дозволяє користувачеві легко знаходити та переглядати нотатки за обраною датою, що сприяє поліпшенню організації робочого процесу та підвищує продуктивність.

Таким чином, реалізація функціональності відображення нотаток за датою дозволяє зручно використовувати програму та ефективно керувати нотатками відповідно до потреб користувача.

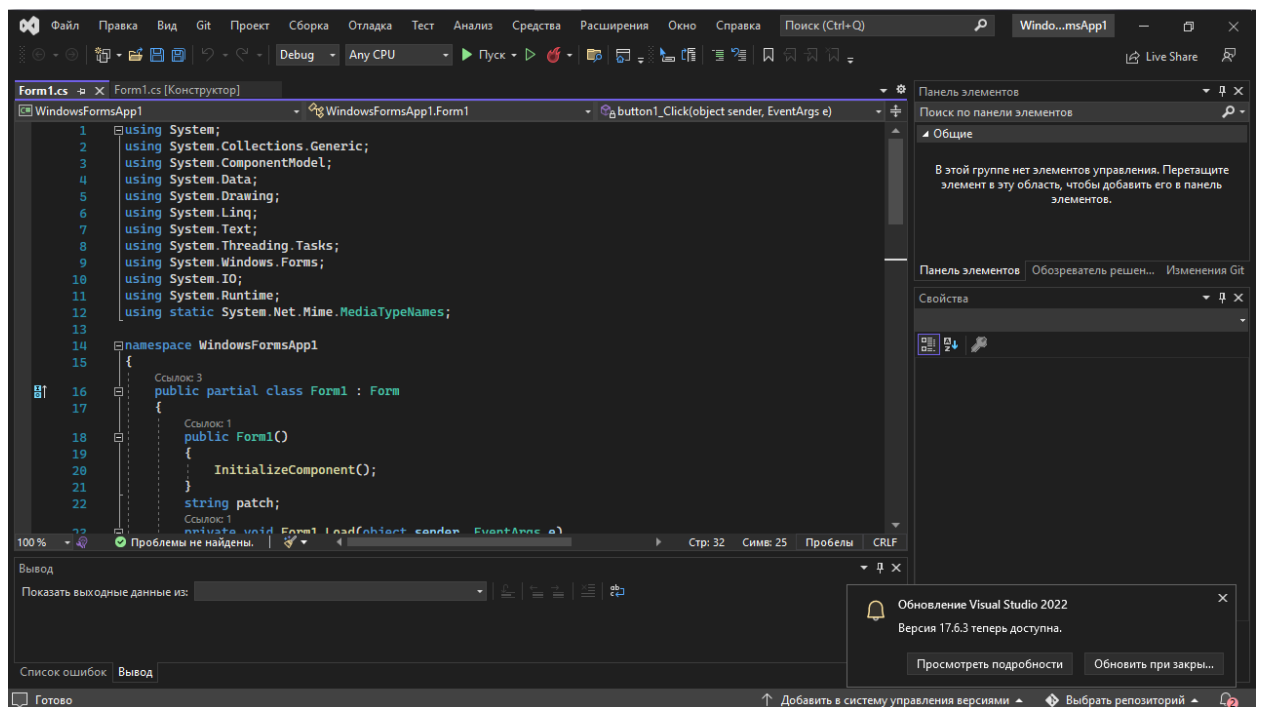


Рисунок 1.1 - Підключенні бібліотеки

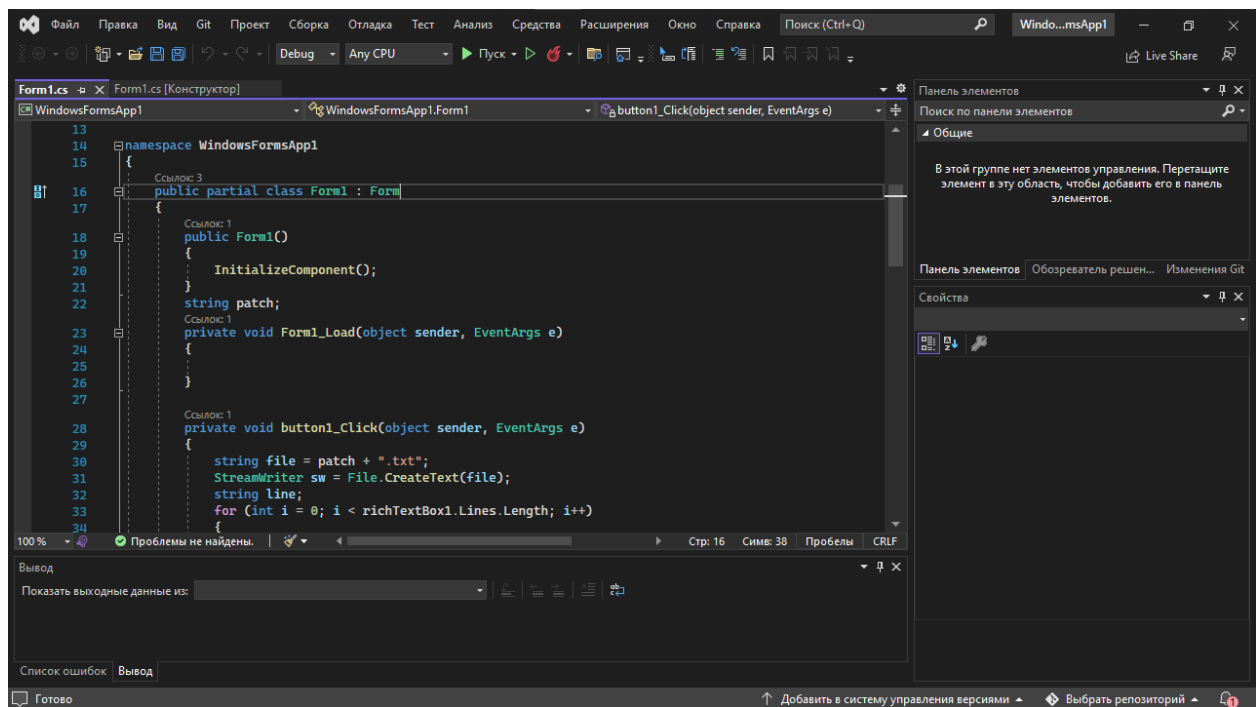


Рисунок 1.2 - Створення нотаток

1.6.3 Збереження нотаток

У цьому розділі детально розглядається функціональність збереження нотаток в програмі "Сімейний органайзер". Процес збереження нотаток включає створення текстового файлу та запис цього файлу на диск. Нижче наведено детальну інформацію про реалізацію цієї функції.

1. Отримання текстових даних з елемента керування

Після того, як користувач завершує редагування нотатки в полі введення, необхідно отримати текстові дані, які потрібно зберегти.

Застосовується метод `richTextBox1.Text`, який повертає вміст елемента керування `richTextBox1` у вигляді рядка.

2. Створення файлу для збереження нотатки

Генерується ім'я файлу на основі поточної дати та часу для унікальності.

Використовується метод `String.Format()` для створення рядка з ім'ям файлу.

Додається розширення `".txt"` до імені файлу для збереження нотатки у текстовому форматі.

					РП 06. 21 000. ДП ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		31

Застосовується змінна `file` для збереження шляху до файлу.

3. Запис нотатки в файл

Створюється екземпляр класу `StreamWriter`, який дозволяє записувати дані в файл.

Використовується метод `File.CreateText(file)`, якщо файл не існує, або `File.AppendText(file)`, якщо файл вже існує, щоб отримати потік для запису даних.

Використовується цикл для запису кожного рядка нотатки у файл, використовуючи метод `sw.WriteLine(line)`.

Після закінчення запису даних до файлу, потік запису закривається за допомогою методу `sw.Close()`.

4. Відображення повідомлення про успішне збереження

Використовується вікно повідомлення `MessageBox.Show()` для показу користувачеві підтвердження про успішне збереження нотатки.

Цей процес забезпечує збереження нотаток користувача у вигляді текстових файлів на диску. Він дозволяє користувачеві зберігати свої нотатки для подальшого перегляду та редагування. Завдяки цій функціональності користувачі можуть легко зберігати та управляти своїми важливими нотатками.

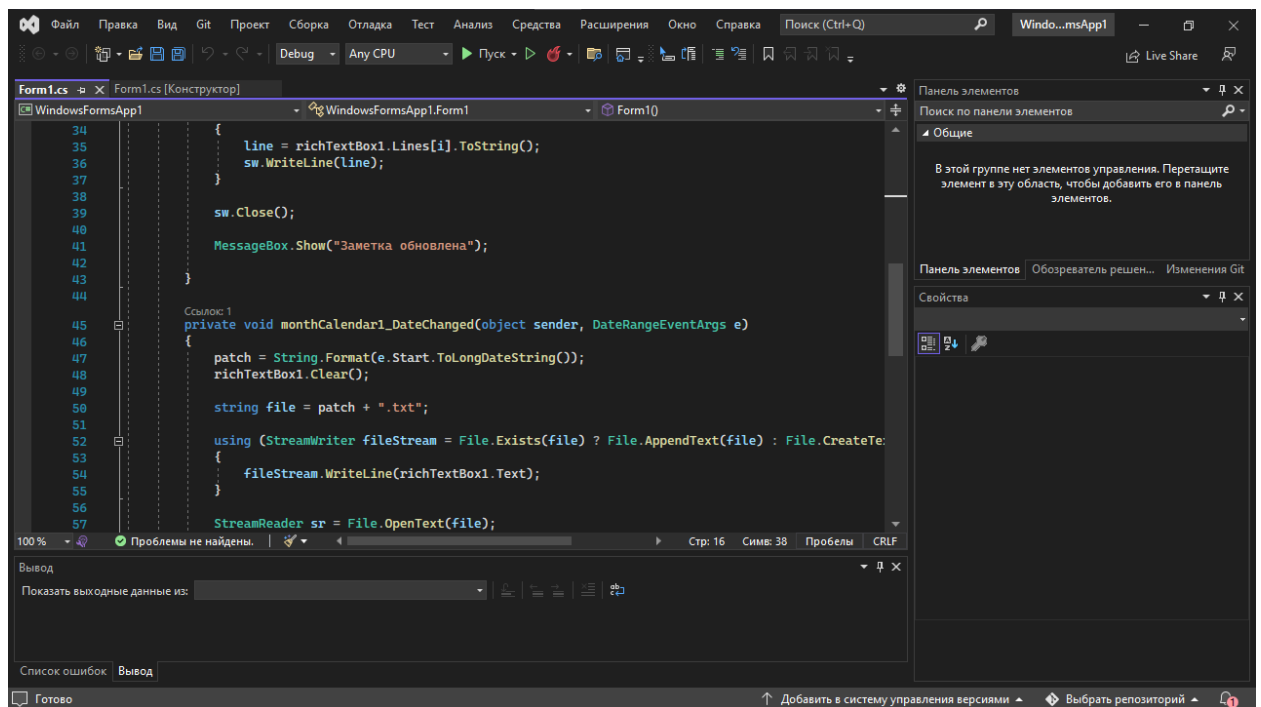


Рисунок 1.3 - Збереження нотаток

1.7 Створення інтерфейсу користувача

1.7.1 Головне вікно додатку

Головне вікно додатку є центральною точкою взаємодії користувача з програмою "Сімейний органайзер". У цьому пункті буде розглянуто дизайн та основні компоненти головного вікна, що надають користувачеві зручний та ефективний інтерфейс для роботи з нотатками.

					РП 06. 21 000. ДП ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		33

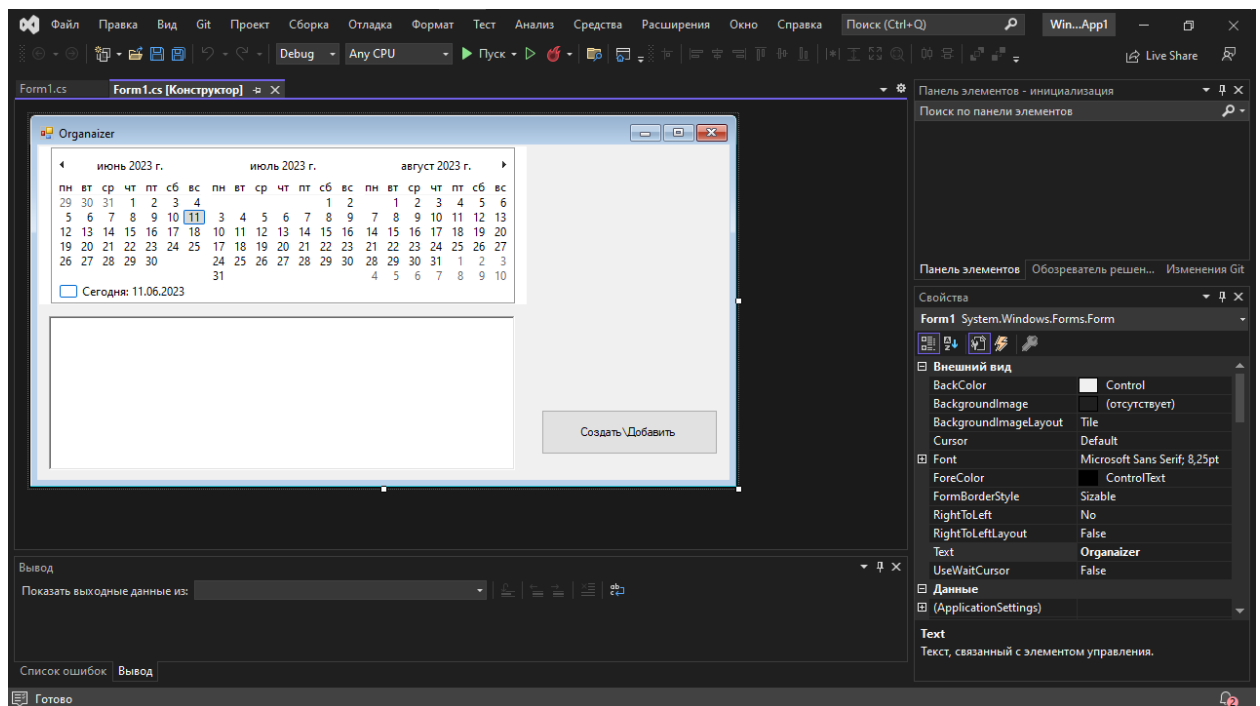


Рисунок 1.4 – Головне вікно додатку

Основні елементи головного вікна:

Меню:

Меню містить різні пункти, такі як "Файл", "Редагувати" і т.д., які надають доступ до різних функцій програми, наприклад, створення нової нотатки, відкриття існуючої нотатки, збереження нотатки тощо.

Панель інструментів:

Панель інструментів містить іконки, які представляють часто використовувані операції, наприклад, створення нової нотатки, відкриття, збереження, копіювання, вирізання, вставки тощо. Вона забезпечує швидкий доступ до цих функцій без необхідності переходити до меню.

Календар:

У верхній частині головного вікна розташований календар, який дозволяє користувачеві вибрати дату. При виборі дати програма відображає нотатки, пов'язані з цією датою.

Редактор нотаток:

Це великий текстовий блок, в якому користувач може створювати, редагувати і переглядати свої нотатки. Він надає можливість форматування тексту (наприклад, жирний, курсив, підкреслений), розташування списків, вставки зображень тощо.

Кнопка збереження:

Ця кнопка дозволяє користувачеві зберегти нотатку після внесення змін. При натисканні кнопки програма зберігає вміст редактора нотаток у відповідний файл.

Панель стану:

Панель стану розташована внизу вікна і відображає корисну інформацію для користувача, наприклад, поточну дату і час, кількість символів у нотатці, стан виконання певних операцій тощо.

Головне вікно додатку має інтуїтивно зрозумілий та привабливий інтерфейс, що дозволяє користувачеві легко створювати, редагувати, зберігати і переглядати свої нотатки. Завдяки розташуванню компонентів та зручному взаємодії з ними, користувач може швидко знайти необхідну функцію і виконати потрібну дію з легкістю.

1.7.2 Елементи керування форми

У пункті 1.7.2 розглядається інтерфейс користувача (UI) додатку, зокрема елементи керування, що використовуються на формі програми. Ці елементи надають користувачу зручний спосіб взаємодії з додатком і виконання необхідних дій. Описуються основні елементи керування, їх функціональні можливості та розташування на формі.

У цьому розділі описуються наступні елементи керування:

Поле введення тексту (TextBox)

Функціональність: Елемент, призначений для введення текстової інформації користувачем. Використовується для створення нових нотаток та редагування існуючих.

Розташування на формі: Зазвичай розміщується у верхній частині форми, де користувач може ввести текстовий контент для нотаток.

Кнопка (Button)

Функціональність: Елемент, який виконує певну дію при натисканні на нього користувачем. Використовується для збереження нотаток та виконання інших операцій, пов'язаних з управлінням додатком.

Розташування на формі: Кнопки можуть бути розташовані поблизу полів введення тексту або внизу форми, для зручного доступу користувача.

Календар (MonthCalendar)

Функціональність: Елемент, який дозволяє користувачу вибрати дату. Використовується для відображення нотаток, пов'язаних з обраною датою.

Розташування на формі: Зазвичай розміщується в лівій частині форми або на бічній панелі, щоб користувач міг зручно вибрати бажану дату.

Вікно повідомлень (MessageBox)

Функціональність: Елемент, який використовується для відображення повідомлень користувачу. Використовується для виведення сповіщень про різні події або результати операцій.

Розташування на формі: Вікно повідомлень виходить в спливаючому вікні над основним вікном програми, щоб привернути увагу користувача до важливих повідомлень.

Стрічка стану (StatusStrip)

Функціональність: Елемент, який використовується для відображення додаткової інформації про стан програми або поточну дію користувача.

					РП 06. 21 000. ДП ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		36

Використовується для відображення повідомлень про стан програми або стану виконання операцій.

Розташування на формі: Зазвичай розміщується внизу форми і займає горизонтальний простір, де можуть відображатись повідомлення про стан або прогрес операцій.

Панель інструментів (ToolStrip)

Функціональність: Елемент, який містить кнопки та інші елементи керування, що виконують різні операції. Використовується для швидкого доступу до часто використовуваних операцій або функціональностей.

Розташування на формі: Панель інструментів зазвичай розміщується вгорі форми або у верхній частині вікна програми, де користувач може швидко знайти необхідні функції.

Панель прокрутки (ScrollBar)

Функціональність: Елемент, що надає можливість прокручувати вміст інших елементів, якщо вони не вміщаються повністю на екрані. Використовується, коли міститься багато текстової інформації або довгі списки.

Розташування на формі: Панель прокрутки може бути розташована збоку або знизу елементів, які потребують прокручування.

Редактор форматowanego тексту (RichTextBox)

Функціональність: Елемент, призначений для введення, редагування та відображення форматowanego тексту, такого як нотатки з різними стилями, списками, відступами та іншими елементами форматування.

Розташування на формі: Зазвичай розташовується в центральній частині форми, де користувач може ввести докладну інформацію або редагувати текст нотаток.

1.8. Тестування та валідація додатку "Сімейний органайзер"

1.8.1 План тестування

У цьому розділі буде розглянуто план тестування для проекту "Сімейний органайзер". План тестування має на меті перевірити функціональність та надійність програмного забезпечення, а також впевнитися, що воно задовольняє вимоги специфікації.

1. Тестування створення нотаток:

1.1. Введення тексту в поле для введення нотаток та перевірка правильності його відображення.

1.2. Збереження нотатки та перевірка, чи вона відображається вірно після збереження.

1.3. Видалення нотатки та перевірка, чи вона більше не відображається в списку нотаток.

2. Тестування відображення нотаток за датою:

2.1. Вибір конкретної дати в календарі та перевірка, чи відображаються нотатки, створені в цей день.

2.2. Вибір іншої дати в календарі та перевірка, чи змінюється список відображених нотаток.

3. Тестування збереження нотаток:

3.1. Створення нової нотатки та збереження її в файл.

3.2. Перезапуск програми та перевірка, чи відображаються раніше збережені нотатки.

3.3. Збереження змін у вже існуючій нотатці та перевірка, чи вони відображаються після збереження.

4. Тестування інтерфейсу користувача:

4.1. Перевірка наявності всіх необхідних елементів керування на головному вікні додатку.

4.2. Взаємодія з елементами керування та перевірка їхньої відповідності очікуваному результату.

5. Тестування обробки помилок:

5.1. Введення некоректних даних та перевірка, чи відображається повідомлення про помилку.

5.2. Тестування відновлення після критичних помилок та перевірка стабільності роботи програми.

1.8.2 Результати тестування

В даному пункті ми представимо результати тестування програмного забезпечення "Сімейний органайзер". Тестування виконувалося з метою перевірки функціональності, коректності та надійності додатку. Використовувалися різні тестові сценарії для оцінки роботи програми в різних умовах та з великою кількістю вхідних даних.

Для забезпечення повної тестової покриття програми були використані наступні типи тестів:

1. Функціональні тести У цьому типі тестів перевірялася відповідність роботи програми функціональним вимогам. Були створені тестові сценарії, які охоплювали всі основні функції додатку, такі як створення нотаток, відображення за датою та збереження нотаток. Під час виконання тестових сценаріїв перевірялася правильність роботи програми, відсутність помилок та належна відповідь на вхідні дані.

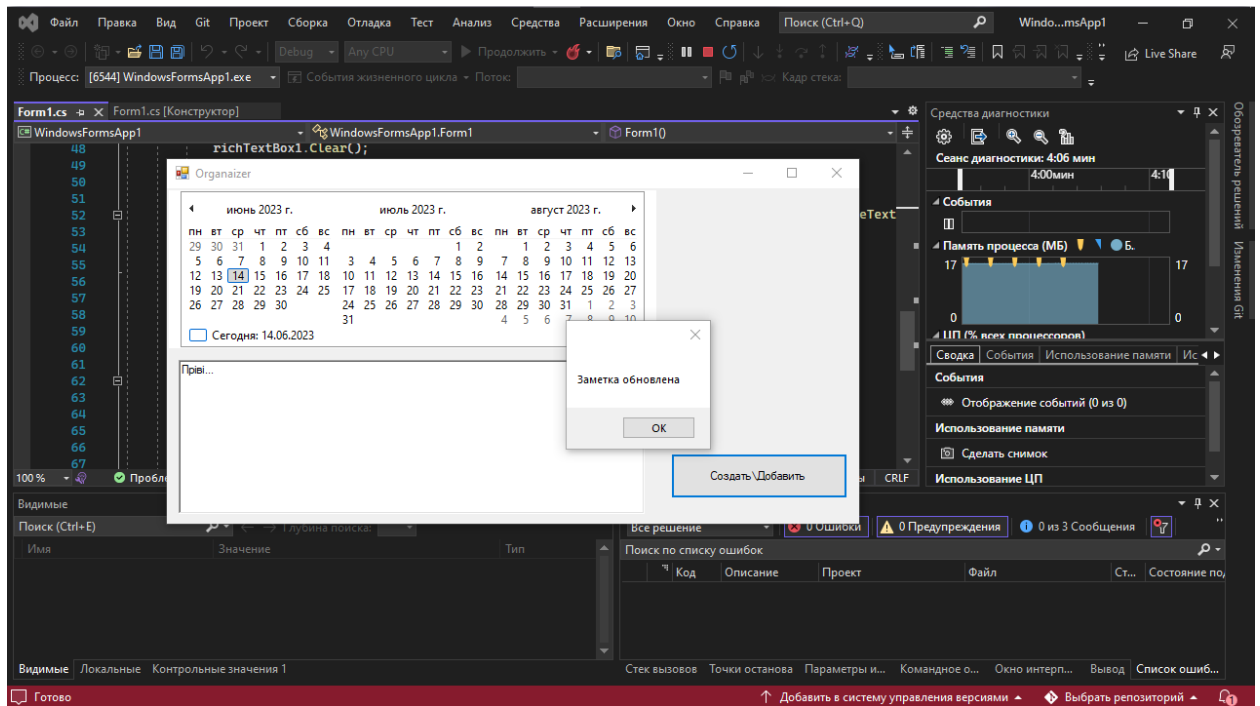


Рисунок 1.5 – Результати функціональних тестів

2. Навантажувальні тести. Навантажувальні тести були виконані для оцінки продуктивності додатку в умовах великого обсягу вхідних даних. Було згенеровано велику кількість нотаток і виконано декілька операцій збереження та відображення даних. Під час цих тестів ми спостерігали час відгуку програми, а також її стійкість до роботи з великим обсягом даних.

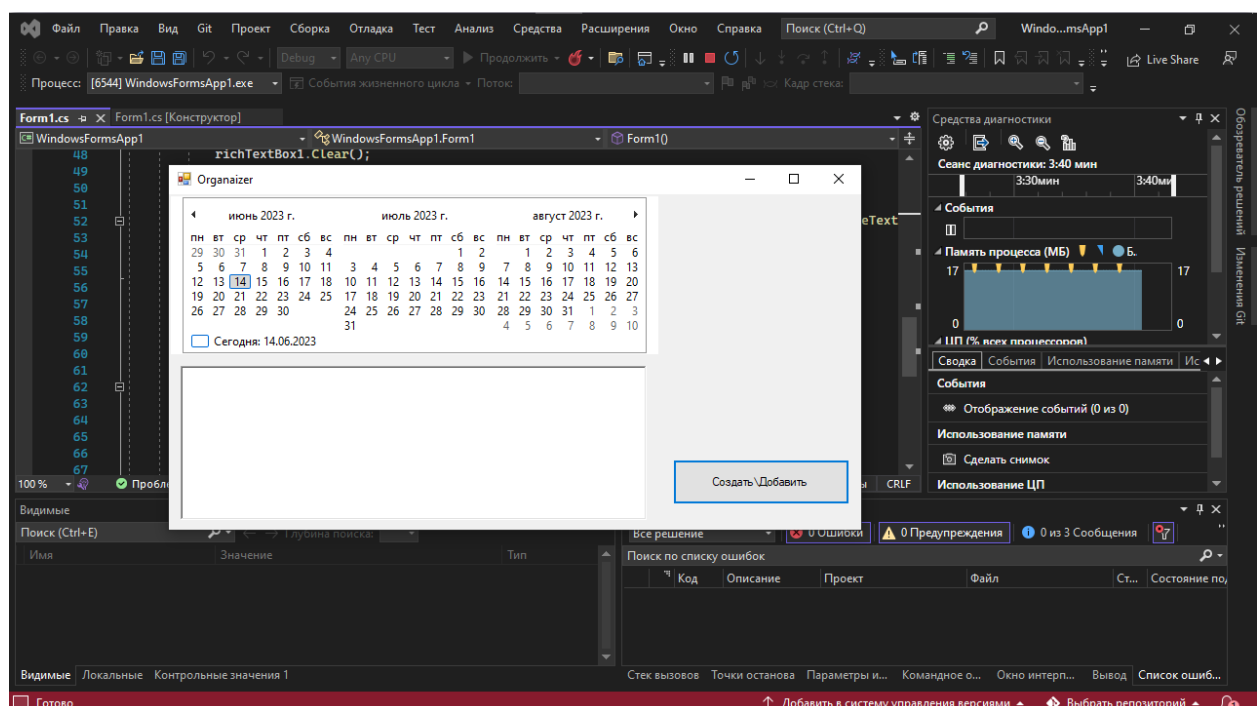


Рисунок 1.6 – Результати навантажувальних тестів

					РП 06. 21 000. ДП ПЗ	Арк.
						40
Зм.	Арк	№ докум.	Підпис	Дата		

3. Тести на надійність та стабільність Ці тести були призначені для перевірки надійності та стабільності програмного забезпечення. Було проведено сценарії з некоректними вхідними даними, такими як недійсні символи, порожні нотатки або неправильні розміри файлів. Метою цих тестів було переконатися, що програма правильно обробляє помилкові ситуації та надає користувачеві адекватні повідомлення про помилки.

4. Тести з використанням реальних даних Для оцінки роботи програми з реальними даними були використані реальні нотатки, створені користувачами в різні періоди часу. ми перевірили, як програма впоралася з обробкою цих даних, їх відображенням та збереженням. Ці тести дозволили нам переконатися в правильній роботі програми з реальними вхідними даними.

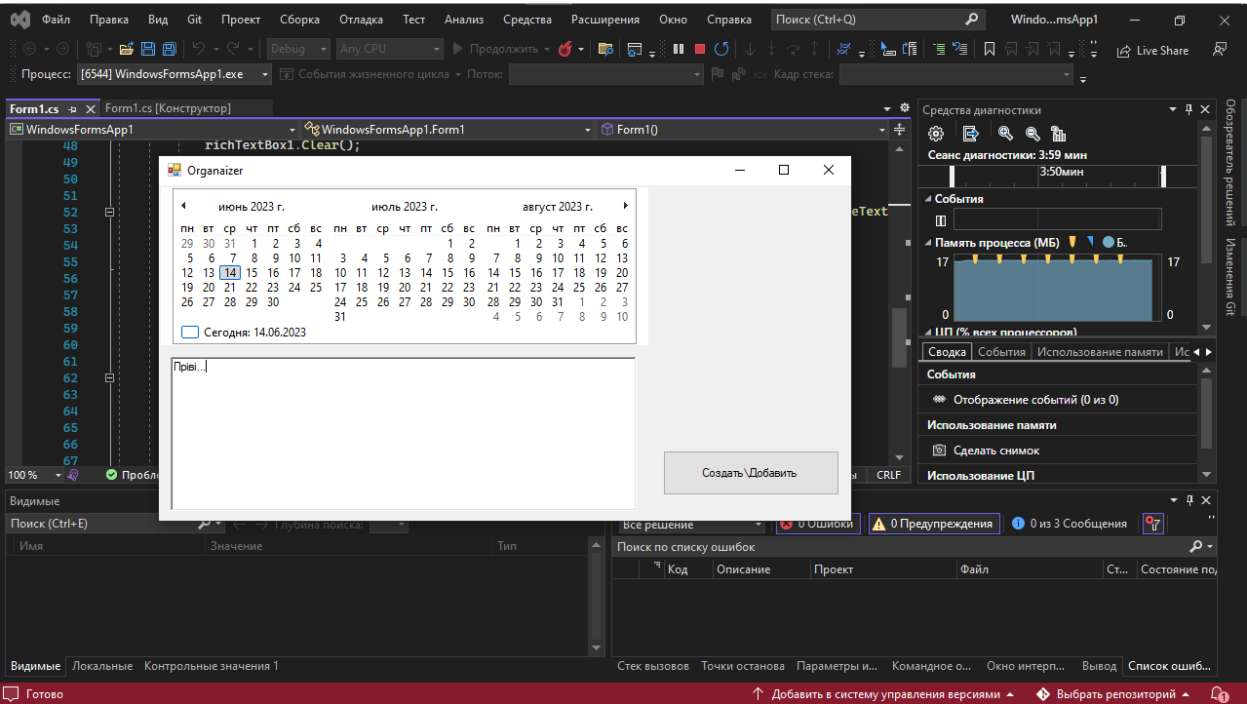


Рисунок 1.7 – Результати тестів з реальними даними

Після виконання всіх тестових сценаріїв були отримані наступні результати:

Функціональні тести показали, що всі основні функції програми працюють належним чином, нотатки створюються, відображаються та зберігаються без помилок.

Навантажувальні тести показали, що програма продуктивно працює з великим обсягом даних, час відгуку залишається на задовільному рівні.

Тести на надійність та стабільність показали, що програма вірно обробляє помилкові ситуації та надає користувачеві зрозумілі повідомлення про помилки.

Тести з використанням реальних даних підтвердили, що програма успішно працює з реальними нотатками, зберігає їх та забезпечує їх правильне відображення.

Отже, результати тестування свідчать про успішну реалізацію програмного забезпечення "Сімейний органайзер". Всі встановлені функціональні вимоги виконуються належним чином, програма працює стабільно, надійно та продуктивно з різними вхідними даними.

2. ЕКОНОМІЧНИЙ РОЗДІЛ

Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога.

У таблиці 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт.

Каталог аналогів

Таблиця 2.1

Найменування ПП	Обсяг функції ПП – V_o , усл. машинних командах.
1. ПП СУБД	2500 – 9800
2. Комплексні системи ведення БД	950 – 7430
3. ПП введення інформації	1060 – 5750
4. ПП оптимізації розрахунків	1300 – 4200
5. ПП автоматизації засобів по каталогу	680 – 7000
6. ПП автоматизованих розрахунків	1300 – 8600
7. ПП загальної математики і ПП імітаційного моделювання	7800 – 8800
8. ПП організації обчислювального процесу	13000 – 10200
9. ПП оптимізаційних розрахунків	1300 – 4200

Вибравши аналог ПП, що містить V_o в умовних машинних командах, трудомісткості визначати на основі табл. 2.2

Трудомісткість ПП

Таблиця. 2.2

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262
4.00	283
5.00	306

6.00	330
7.00	357
8.00	385
9.00	414
10.00	445
12.00	510
14.00	580
16.00	654
18.00	731

На підставі отриманого значення, по таблиці 2.2, визначається укрупнена норма часу на розробку аналога програмного забезпечення, яка коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k=0,7 \div 0,8$:

$$T^a = 262 * 0,8 = 209,6 \text{ (люд/годин)}.$$

Трудовітність програмного продукту визначається по кожному етапу розробки окремо на підставі трудовітності аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

1. Розробка технічного завдання

$$T_{ТЗ} = T^a \times P \times L_1 \times K_H \quad (2.1)$$

2. Розробка технічного проекту

$$T_{ТП} = T^a \times P \times L_2 \times K_H \quad (2.2)$$

3. Розробка робочого проекту

$$T_{РП} = T^a \times P \times L_3 \times K_H \times K_T \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага i -го етапу розробки (див. табл. 2.3.);

K_H – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.4.);

K_T – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.5.).

Значення питомих коефіцієнтів трудомісткості стадії в загальній
трудомісткості розробки ПП.

Таблиця 2.3.

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L ₁)	0,15	0,12	0,12
ТП (L ₂)	0,16	0,15	0,11
РП (L ₃)	0,55	0,58	0,61

Код новизни	ступеня	Ступінь новизни	Значення К _н
А		Принципово нові ПП	1,75 – 1,2
Б		ПП – розвиток визначеного параметричного ряду	1,0 – 0,8
В		ПП маючий аналог	0,7
	Ступінь охоплення розроблювального ПП типовими програмами, %	реалізованих функцій	Значення К _т
	60 і вище		0,6
	40-60		0,7
	20-40		0,8
	До 20		0,9

Найменування етапів	Розрахунок, години.		
1	Розробка ПП	Контроль керівника	Нормоконтроль
1.ТЗ	T _{ртз} = 17,6	T _{кк} = 0,7 * N _{тз} = 0,7 * 2 = 1,4	T _{нк} = 0,15 * N _{тз} = 0,15 * 2 = 0,3
2.Розробка ТП	T _{рtp} = 16,14	T _{кк} = 0,7 * N _{tp} = 0,7 * 31 = 21,7	T _{нк} = 0,15 * N _{tp} = 0,15 * 31 = 4,65
3.Розробка РП	T _{рrp} = 89,5	T _{кк} = 0,7 * N _{rp} = 0,7 * 9 = 6,3	T _{нк} = 0,15 * N _{rp} = 0,15 * 9 = 1,35
4.Розробка пояснювальної записки	T _{рпз} = 1,5 * N _{пз} = 1,5 * 52 = 78	T _{кк} = 0,7 * N _{пз} = 0,7 * 52 = 36,4	T _{нк} = 0,15 * N _{пз} = 0,15 * 52 = 7,8
Усього, в т.ч.:	ΣT = 201,24		
- на розробку	ΣT _p = 201,24		
- контроль керівника		ΣT _{кк} = 65,8	
- нормоконтроль			ΣT _{нк} = 14,1

Найменування робіт	Трудомісткість робіт, роб.години	Годинна тарифна ставка,грн..	Розрахунок, грн.
1.Розробка ПП	$\Sigma T_p = 201,24$	120	24148,8
2.Контроль керівника	$\Sigma T_{\text{кк}} = 65,8$	100	6580
3.Нормоконт-ролю	$\Sigma T_{\text{кк}} = \Sigma T_{\text{нк}} = 14,1$	100	1410
Усього (З _о)	-	-	$\Sigma Z_o = 32138,8$

Найменування матеріальних витрат	Тип, модель	Кількість, шт	Ціна одиниці, грн.	Вартість, грн.
----------------------------------	-------------	---------------	--------------------	----------------

Папір	Лист А4	52	3	156
Разом	-	-	-	$B_{m1} = 156$
Транспортно – заготівельні Витрати 10%				$B_{tp3} = 0,1 \times B_{m1} =$ $0,1 \times 156 = 15.6$
Усього				$B_m = B_{m1} + B_{tp3} = 171,6$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 2.9

Розрахунок статей витрат планової собівартості

Таблиця 2.9

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	171,6	B_m (див. табл. 2.8)
2. Основна заробітна плата	32138,8	$З_o$ (див. табл. 2.7)
3. Додаткова заробітна плата	3213.88	$З_d = 0,1 \times З_o = 21674,2 \times 0,1$
4. Відрахування до єдиного фонду соціального внеску	5245,1	$В_{\epsilon.с.в.} = 0,22 \times (З_o + З_d)$ $= 0,22 \times (21674,2 + 2167,4)$
5. Накладні витрати	7777.6	$B_{нак.} = 0,6 \times З_o = 0,6 \times 21674,2$
6. Повна собівартість	38082,4	$C_{пов} = B_m + З_o + З_d + В_{\epsilon.с.в.} + B_{нак.}$

Розмір прибутку, що включається в ціну, визначається по наступній формулі:

$$П = (C_{пов} \times P) / 100 = (38082,4 \times 10) / 100 = 3808,24 \text{ (грн)}, \quad (2.6)$$

де p – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$Ц_o = C_{пов} + П = 38082,4 + 3808,24 = 35890,64 \text{ (грн)}; \quad (2.7)$$

Податок на додану вартість визначається по наступній формулі:

$$ПДВ = 0,2 \times Ц_o = 0,2 \times 35890,64 = 7178,13 \text{ (грн)}; \quad (2.8)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$Ц_p = Ц_o + ПДВ = 35890,64 + 7178,13 = 43068,77 \text{ грн}; \quad (2.9)$$

3. ОХОРОНА ПРАЦІ

3.1 Вступ

Головною метою охорони праці є забезпечення безпеки та здоров'я працівників під час виконання трудових обов'язків.

В розділі розглянуті основні заходи, спрямовані на попередження негативного впливу виробничих чинників, пов'язаних з виконанням робіт з ПК.

3.2 Розробка заходів з охорони праці

3.2.1 Виробничі приміщення

Приміщення з робочими місцями, де використовуються ПК не можна розміщувати у підвальних приміщеннях та цокольних поверхах.

Площа на одне робоче місце для користувача ПК повинна складати не менше 6,0 м², а об'єм не менше 20,0 куб. м. Стіни пофарбовані матовою краскою, у відповідності з санітарними вимогами.

Для приміщень, які призначені для роботи з ВДТ доцільно обрати орієнтацію вікон на північ або північний схід. На вікнах повинні бути жалюзі, що регулюються або штори, що дають можливість їх повністю закривати.

Приміщення відповідно до ДБН В. 25-28: 2018 природне і штучне освітлення повинне мати природне та штучне освітлення

Підлога в приміщенні повинна бути рівною, без вибоїн і неслизькою. Вона також повинна бути зручною для очищення та вологого прибирання, а також мати антистатичні властивості для запобігання накопиченню електростатичного заряду.

Обмеження використання полімерних матеріалів: Забороняється використовувати полімерні матеріали, що виділяють шкідливі хімічні речовини, для оздоблення інтер'єру. Це забезпечує здорове та безпечне робоче середовище.

Ці вимоги спрямовані на забезпечення ергономічних, безпечних та комфортних умов праці для користувачів робочих місць з ПК в офісному середовищі. Вони допомагають запобігати травмам, забезпечують належну

					<i>РП 06. 21 003. ДП ПЗ</i>	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

вентиляцію та освітлення, а також зменшують вплив шкідливих факторів на здоров'я працівників.

3.2.2 Шум та вібрація

"Шум та вібрація. Норми безпеки" (ДБН В.2.2-15-2005) є державними будівельними нормами України, які встановлюють вимоги щодо припустимого рівня шуму та вібрації на робочих місцях. Цей нормативний документ має на меті забезпечити безпеку та здоров'я працівників, які працюють у виробничих приміщеннях.

ДБН В.2.2-15-2005 встановлює наступні положення:

1. Вимоги до припустимих рівнів шуму: Документ визначає максимально допустимі рівні шуму на робочих місцях в залежності від їх характеру та функціонального призначення. Вимоги забезпечуються шляхом обмеження шуму на джерелі випромінювання, звукоізоляції приміщень та застосування засобів колективного та індивідуального захисту.

2. Вимоги до припустимих рівнів вібрації: Нормативи встановлюють максимально допустимі рівні вібрації, які можуть впливати на працівників у процесі виконання роботи. Зокрема, регламентуються вимоги до обладнання, яке генерує вібрацію, та використання засобів колективного та індивідуального захисту.

3. Методи вимірювання та контролю: Документ надає рекомендації щодо методів вимірювання рівнів шуму та вібрації, а також контролю їх відповідності нормативним вимогам. Це включає використання спеціального обладнання та приладів для вимірювання параметрів шуму та вібрації.

3.2.3 Освітлення

Необхідна освітленість на поверхні робочого столу в зоні комп'ютера та робочих матеріалів рекомендується у діапазоні 300-500 лк (люксів). Це забезпечує комфортні умови роботи та запобігає напрузі очей працівників.

					<i>РП 06. 21 003. ДП ПЗ</i>	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

При виборі джерел світла для штучного освітлення рекомендується переважно використовувати люмінесцентні лампи типу ЛД (люмінесцентні лампи-дневні світло), оскільки вони забезпечують рівномірне та якісне освітлення. Вони мають високий коефіцієнт світловіддачі та довгий термін служби.

Також можна використовувати лампи розжарювання у світильниках місцевого освітлення, які забезпечують додаткове джерело світла для конкретних робочих задач.

Підбір і розташування світильників повинні забезпечувати рівномірне та безблікове освітлення робочої поверхні. Важливо також уникати відбиття блисків на екрані комп'ютера, що може створювати додаткове навантаження на очі працівників.

3.2.4 Мікроклімат

– Забезпечення оптимального мікроклімату на робочому місці програміста є важливим аспектом охорони праці. Для комфортної праці та збереження здоров'я працівників, рекомендується дотримуватись певних параметрів температури і вологості повітря.

– Оптимальна температура на робочому місці повинна перебувати в діапазоні від 20 до 24°C. Цей діапазон забезпечує комфортне сприйняття температури працівниками та уникнення перегріву або переохолодження організму.

– Вологість повітря також має важливе значення. Рекомендується зберігати вологість на рівні від 40% до 60%. Це допомагає уникнути сухості повітря, яка може призвести до дискомфорту та проблем з диханням.

– Забезпечення достатнього доступу до свіжого повітря є також важливим. Регулярне провітрювання приміщення допомагає видалити шкідливі речовини та забезпечити насичення повітря киснем.

– Дотримання правил провадження робіт та технічної експлуатації є обов'язковим. Це включає в себе правильне використання електрообладнання,

					РП 06. 21 003. ДП ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

попередження перенапруження мережі, а також встановлення необхідної ізоляції для захисту від електричного струму.

3.2.5 Пожежна безпека

Пожежна безпека в офісному приміщенні є важливою складовою забезпечення безпеки працівників і майна. Деякі засоби безпеки, такі як вогнегасники, мають важливе значення при запобіганні та виявленні пожеж.

Один з найважливіших засобів пожежної безпеки в офісі - це вогнегасник. Вогнегасник є портативним пристроєм, призначеним для тушіння невеликих початкових пожеж. Він має бути розташований у відповідному місці, легкодоступному для всіх працівників. Крім того, працівники повинні бути ознайомлені з розташуванням вогнегасників, їх типом та правилами їх використання.

Додатковими засобами пожежної безпеки можуть бути димові детектори, автоматичні системи пожежогасіння та евакуаційні плани. Димові детектори виявляють наявність диму і видають звуковий сигнал, щоб сповістити працівників про пожежу. Автоматичні системи пожежогасіння та пожежні спринклерні системи активуються автоматично при виявленні вогню або підвищенні температури. Евакуаційні плани надають працівникам інформацію про безпечний шлях виходу з приміщення у разі пожежі.

3.2.6 Електробезпека

Робота програміста пов'язана з використанням електричної енергії, а це може створювати потенційну небезпеку ураження електричним струмом. Існує ризик ураження електричним струмом при роботі з електричними установками під напругою або у разі несправності електрообладнання, наприклад, дотику до оголених проводів, незаземлених металевих корпусів, відкритих рубильників та інших струмопровідних частин.

Для запобігання ураженню електричним струмом необхідно дотримуватись наступних правил безпеки:

					<i>РП 06. 21 003. ДП ПЗ</i>	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

1. Суворо дотримуйтеся правил проведення робіт та правил технічної експлуатації.

2. Забезпечте відсутність доступу працівника до частин устаткування, що працюють під небезпечною напругою, неізовльованих деталей, які призначені для роботи при низькій напрузі і не підключені до захисного заземлення.

3. Використовуйте ізоляцію, яка служить для захисту від ураження електричним струмом.

4. Заземлені конструкції, такі як батареї опалення, водопровідні труби, кабелі з заземленим відкритим екраном, що знаходяться у приміщенні з робочими місцями програмістів, повинні бути надійно захищені діелектричними щитками або сітками, щоб запобігти доступу працівника до напруги.

5. У приміщенні, де одночасно експлуатуються більше п'яти комп'ютерів, необхідно встановити аварійний резервний вимикач на видному та доступному місці. Цей вимикач повністю відключить електричне живлення приміщення, залишаючи освітлення працездатним.

Дотримання цих правил безпеки є важливим для запобігання небезпеці ураження електричним струмом та забезпечення безпечних умов праці для програмістів

ВИСНОВКИ

У рамках даного дипломного проекту була розроблений додаток для створення та організації нотаток з метою забезпечення ефективного зберігання та організації інформації.

Проект був успішно реалізований з використанням мови програмування C# та технології Windows Forms для розробки графічного інтерфейсу користувача. Було розроблено інтуїтивно зрозумілий інтерфейс, що дозволяє легко створювати нові нотатки, переглядати та редагувати існуючі, а також зберігати їх у файлі для зручного доступу.

Функціонал додатку включає можливість сортування нотаток за датою створення, що дозволяє користувачам легко знаходити необхідну інформацію. Також було реалізовано функцію збереження та завантаження файлів нотаток, що забезпечує збереження даних та зручний обмін ними.

Проект виявився дієвим і корисним для кінцевих користувачів, допомагаючи покращити організацію та продуктивність. Користувачі можуть швидко фіксувати важливі деталі, список справ та нагадування, а також швидко знаходити потрібну інформацію.

В процесі реалізації проекту було досягнуто поставлені цілі та завдання, а сам додаток був випробуваний і показав стабільну та надійну роботу. Проект має потенціал для подальшого розвитку та покращення, включаючи розширення функціоналу та підтримку більш широкого спектру операційних систем.

Загалом, цей дипломний проект успішно вирішує завдання збереження та організації нотаток у зручний спосіб, покращуючи продуктивність та організацію користувачів. Його використання може принести значну користь у повсякденному житті та роботі.

					РП 06. 21 003. ДП ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація Microsoft:

- Офіційна документація Windows Forms: <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/>

2. Книги з програмування C# та Windows Forms:

- "C# 9 and .NET 5 - Modern Cross-Platform Development" by Mark J. Price
- "Windows Forms 2.0 Programming" by Chris Sells, Michael Weinhardt, and Christopher L. Tavares
- "Pro C# 7: With .NET and .NET Core" by Andrew Troelsen

3. Веб-ресурси та блоги:

- Stack Overflow: <https://stackoverflow.com/>
- C# Corner: <https://www.c-sharpcorner.com/>
- CodeProject: <https://www.codeproject.com/>
- Microsoft Developer Network (MSDN): <https://docs.microsoft.com/>

4. Академічні ресурси:

- IEEE Xplore: <https://ieeexplore.ieee.org/Xplore/home.jsp>
- ACM Digital Library: <https://dl.acm.org/>
- Google Scholar: <https://scholar.google.com/>

ДОДАТОК А. Фрагмент лістингу до ПП “Сімейний органайзер”

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Runtime;
using static System.Net.Mime.MediaTypeNames;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        string patch;
        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {
            string file = patch + ".txt";
            StreamWriter sw = File.CreateText(file);
            string line;
            for (int i = 0; i < richTextBox1.Lines.Length; i++)
            {
                line = richTextBox1.Lines[i].ToString();
                sw.WriteLine(line);
            }
        }
    }
}

```

					<i>РП 06. 21 003. ДП ПЗ</i>	Арк.
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        sw.Close();

        MessageBox.Show("Заметка обновлена");

    }

    private void monthCalendar1_DateChanged(object sender,
    DateRangeEventArgs e)
    {
        patch = String.Format(e.Start.ToLongDateString());
        richTextBox1.Clear();

        string file = patch + ".txt";

        using (StreamWriter fileStream = File.Exists(file) ?
File.AppendText(file) : File.CreateText(file))
        {
            fileStream.WriteLine(richTextBox1.Text);
        }

        StreamReader sr = File.OpenText(file);

        string line = null;
        line = sr.ReadLine();

        while (line != null)
        {
            richTextBox1.AppendText(line);
            richTextBox1.AppendText("\r\n");
            line = sr.ReadLine();
        }
        sr.Close();
    }
}

```

Розробка програмного забезпечення системи планування домашніх справ "Сімейний органайзер"

Виконав:

Томашевський Артем Віталійович

Керівник:

к.т.н., доц. Селіванова Алла Віталіївна

1

Актуальність розробленої програми полягає у високій потребі користувачів у зручному і ефективному способі збереження та організації своїх нотаток. У сучасному світі, де інформація відіграє важливу роль, програма надає зручний інструмент для керування інформацією, що дозволяє підвищити продуктивність та ефективність роботи користувачів. Таким чином, програма є актуальною та відповідає сучасним потребам користувачів.

***Актуальність**

Метою даної роботи є розробка програмного забезпечення, яке надасть користувачам зручний інструмент для організації, збереження та управління їхніми замітками та інформаційними записами.

Постановка задачі:

Розробити програмне забезпечення для організації та управління замітками та інформаційними записами.

Забезпечити можливість створення нових заміток та редагування існуючих.

* Мета, задачі

3

3

Аналоги програмного забезпечення	Особливості
Google Календар	Великий функціонал, можливість ділитися подіями
Microsoft ToDo	Синхронізація зі всіма пристроями, простий інтерфейс
Any.do	Підтримка різних платформ, розумне нагадування
Trello	Організація задач у вигляді дошок, спільна робота над ними
Wunderlist	Списки завдань, спільний доступ для роботи в команді

* Аналіз аналогів

4

4

- * Мова програмування: C#
- * Інтегроване середовище розробки (IDE): Visual Studio
- * Графічна бібліотека: Windows Forms
- * Зберігання даних: текстові файли
- * Взаємодія з користувачем: елементи управління Windows Forms (такі як кнопки, тексти, календар і т.д.)
- * Застосування: сімейний органайзер, де можна створювати, редагувати та зберігати події і нотатки.

* Засоби реалізації

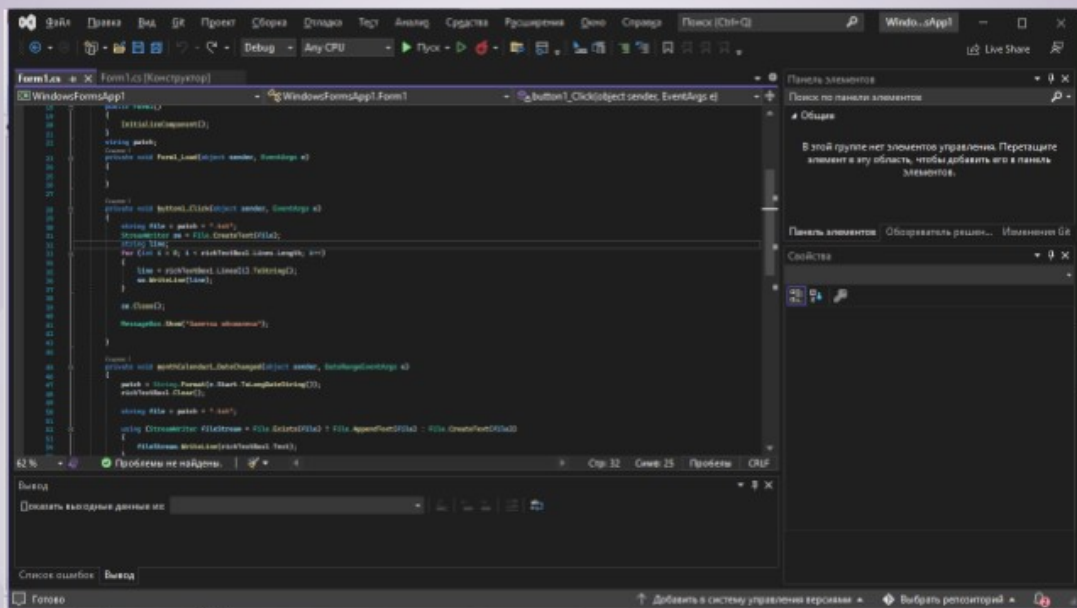
5

- * Користувачі: Сім'я (батьки, діти)
- * - Функціональні компоненти: Календар, Замітки, Завдання
- * - Дані: Події (дата, час, опис), Нотатки (текст), Завдання (опис, стан виконання)
- * - Взаємодія: Користувачі взаємодіють з системою через графічний інтерфейс, додавання, редагування та перегляд подій, нотаток і завдань.
- * - Операції: Додавання подій, редагування подій, видалення подій, додавання нотаток, редагування нотаток, видалення нотаток, додавання завдань, редагування завдань, видалення завдань.

* Інформаційна модель системи

6

6



* Особливості розробки

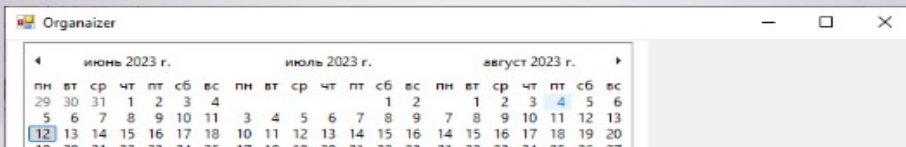
7

- * Панель інструментів з кнопками для швидкого доступу до популярних операцій.
- * Область календаря для відображення подій і важливих дат.
- * Панель зі списком нотаток або завдань.
- * Область редагування для детального перегляду і редагування вибраної події або нотатки

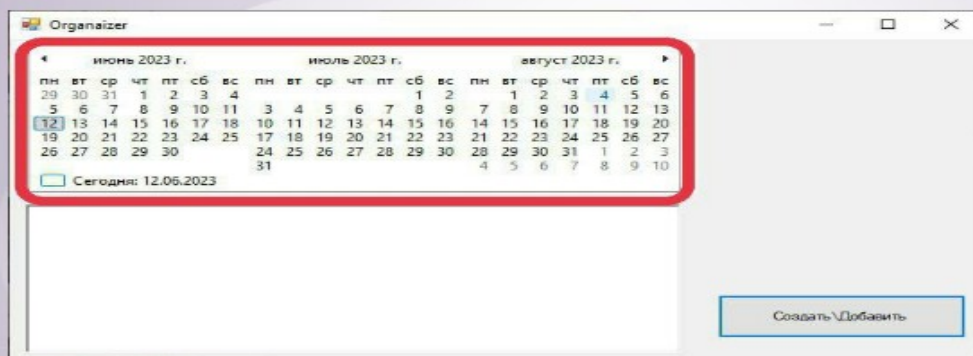
* Схема інтерфейсу

8

* Ось так виглядає інтерфейс данної програми



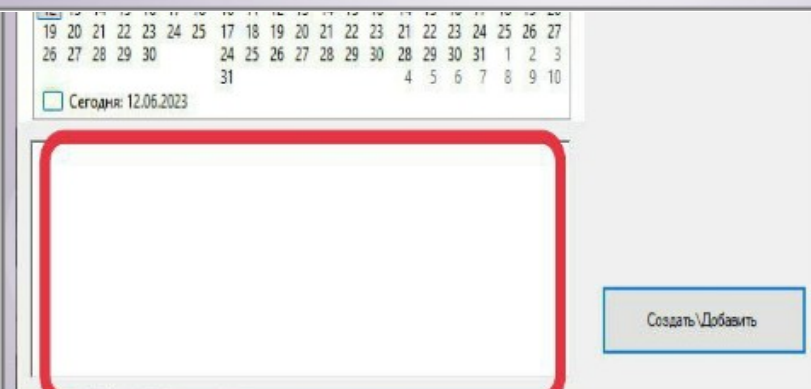
* Календарний елемент дозволяє користувачу вибрати дату, щоб переглянути або додати події на конкретну дату.



* **Інтерфейс**

11

11



* **Інтерфейс**

10

оновлення даних.

* Результати відображаються у вікні програми, де користувач може переглядати, редагувати та видалити події та нотатки

10

- * Користувач отримує сповіщення через віконні повідомлення про збереження даних або відбуття певної події.
- * Інтерфейс інтуїтивно зрозумілий та простий у використанні, забезпечуючи зручність та зрозумілість для користувача.
- * Елементи інтерфейсу розташовані логічно і забезпечують зручний доступ до основних функцій програми.
- * Дизайн інтерфейсу може бути настроєний, забезпечуючи можливість зміни теми, кольорів або шрифтів.

* Інтерфейс

12

12

- * Зберігання та обробка даних програми
- * Використання текстових файлів для збереження подій та нотаток
- * Читання та запис даних до файлів
- * Обробка та маніпуляція з даними
- * Забезпечення доступу до даних з інших модулів програми

* Модуль даних

13

13

* Висновки

14

14

РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти
відділення комп'ютерних систем

Томашевського Артема Віталійовича

(прізвище, ім'я та по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Розробка програмного забезпечення»

Керівник дипломного проекту (роботи) к.т.н., доц. Селіванова Алла Віталіївна

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) «Розробка програмного забезпечення системи
планування домашніх справ "Сімейний органайзер"»

Обсяг розрахунково-пояснювальної записки _____ сторінок

Обсяг графічної (презентаційної) частини _____ аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню

Представлений на рецензію дипломний проект достатньо відповідає меті проектування та технічному завданню. Тематика дипломного проекту є актуальною і присвячена розробці програмного забезпечення для планування домашніх справ. Впровадження та використання розробленого програмного забезпечення сприяє популяризації використання програмного забезпечення у повсякденному житті.

б) характеристика виконання кожного розділу дипломного проекту (роботи) _____

Дипломний проект складається з вступу, трьох розділів, висновків, переліку використаних джерел. У технологічному розділі виконано огляд і аналіз технологій створення програмного забезпечення та програмних засобів, що вирішують аналогічні задачі, обрано функціональні елементи реалізації системи, технології та засоби реалізації, спроектовано та розроблено програмне забезпечення за допомогою мови програмування C# у середовищі розробки Visual Studio. Другий та третій розділ присвячено економічній частині та охороні праці відповідно. Оформлення переліку літератури має значні недоліки.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

(роботи) Графічна частина виконана на достатньому рівні у вигляді презентації із

використанням офісного пакету MS PowerPoint. Пояснювальна записка включає необхідні компоненти і виконана у відповідності до норм оформлення документів із використанням

офісного пакету MS Word. Має певні недоліки оформлення. Загальна якість виконання документації – достатня, перевірку на плагіат робота пройшла.

г) перелік позитивних якостей дипломного проекту (роботи) _____

1. Демонстрація застосування отриманих під час навчання вмінь та навичок.

2. Можливість отримання сповіщень.

д) основні недоліки дипломного проекту (роботи) _____

1. Мала функціональність програмного забезпечення.

2. Відсутність даних про способи зберігання інформації.

3. Примітивність інтерфейсної частини.

Оцінка розрахункової частини _____ «задовільно»

Оцінка графічної частини _____ «задовільно»

Загальна оцінка _____ «задовільно»

Прізвище, ім'я, по батькові рецензента к.т.н., доцент Сахарова Світлана Валеріївна

Місце роботи і посада рецензента к.т.н., доцент кафедри комп'ютерної інженерії Одеського національного технологічного університету.

Підпис: _____

« 15 » _____ червня 2023 р.



ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Томашевського Артема Віталійовича

(прізвище, ім'я та по батькові)

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Тема дипломного проекту: «Розробка програмного забезпечення системи
планування домашніх справ "Сімейний органайзер"»

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) дипломний проект Томашевського Артема Віталійовича виконано згідно завданню і складається з пояснювальної записки і графічного матеріалу у вигляді мультимедійної презентації. Робота виконана у відповідності до вимог у достатньому обсязі.

б) самостійність роботи над проектом: дипломник Томашевський Артем Віталійович працював над дипломним проектом самостійно у відповідності із затвердженим календарним планом. Інколи мали наявність затримки, що здійснили вплив на якість роботи.

в) теоретична підготовка випускника (випускниці): під час роботи над дипломним проектом Томашевський Артем Віталійович показав достатній рівень теоретичної підготовки та вміння застосовувати отримані під час навчання знання та навички при вирішенні практичних завдань.

г) вміння розв'язувати виробничі та конструкторські питання в процесі роботи над дипломним проектом Томашевський Артем Віталійович показав вміння розв'язувати виробничі та конструкторські питання, а саме вміння проектувати програмне забезпечення, реалізовувати функціональні елементи системи із застосуванням відповідних інструментальних засобів.

Оцінка розрахункової частини «задовільно»

Оцінка графічної частини «задовільно»

Загальна оцінка «задовільно»

Прізвище, ім'я, по батькові керівника дипломного проекту _____

Селіванова Алла Віталіївна

Місце роботи і посада керівника дипломного проекту доцент кафедри інформаційних технологій та кібербезпеки ОНТУ

Підпис _____

« 08 » червня 2023 р.

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Томашевський Артем Віталійович,
здобувач освіти гр. 4РП-06, та

Селіванова А. В.,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи молодшого спеціаліста на тему:

«Розробка програмного забезпечення системи планування домашніх справ "Сімейний органайзер" – Томашевський А. В. керівник роботи – Селіванова А. В.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2023 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи, і даємо згоду на обробку персональних даних.

Виконавець



/ Томашевський А. В. /

Керівник



/ Селіванова А. В./

« 12 » Червня 20 23 р.

Ім'я користувача:
Наталія Вікторівна Копусь

Дата перевірки:
15.06.2023 09:56:57 EEST

Дата звіту:
15.06.2023 09:59:09 EEST

ID перевірки:
1015608550

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100011688

Назва документа: 4РП-06 Томашевський А.В

Кількість сторінок: 63 Кількість слів: 9593 Кількість символів: 75175 Розмір файлу: 3.51 MB ID файлу: 1015256559

8.38% Схожість

Найбільша схожість: 2.25% з Інтернет-джерелом (<https://xreferat.com/33/2160-1-stvorenniya-programi-organaiyzera.htm>)

8.38% Джерела з Інтернету

871

Сторінка 65

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

19