

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕКОНОМІКИ ТА МЕНЕДЖМЕНТУ**

Кафедра кібернетики та інформатики

Кваліфікаційна робота

освітній ступінь – «Бакалавр»

**на тему: Розробка інформаційної підсистеми
матеріального забезпечення інформаційної системи
коледжу**

Виконав:

студент спеціальності

126 «Інформаційні системи та технології»

ОП «Інформаційні системи та технології»

Шакотько Артур Михайлович

Керівник:

Толбатов Андрій Володимирович

доцент кафедри кібернетики та інформатики

Рецензент:

Гриценко Костянтин Григорович

доцент кафедри економічної кібернетики
ННІ бізнесу, економіки та менеджменту
Сумського державного університету

Суми – 2023

СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

Факультет	Економіки і менеджменту
Кафедра	Кібернетики та інформатики
Освітній ступінь	«Бакалавр»
Спеціальність	126 «Інформаційні системи та технології» ОП «Інформаційні системи та технології»

Затверджую:
Завідувач кафедри Світлана АГАДЖАНОВА

«_____» «_____» 2023 ____ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ *Шакотько Артуру Михайловичу*

1. Тема роботи: Розробка інформаційної підсистеми матеріального забезпечення інформаційної системи коледжу ВСП «ГАФК СНАУ»

2. База практичного дослідження: Відокремлений структурний підрозділ «Глухівський агротехнічний фаховий коледж Сумського НАУ»

керівник роботи Толбатов Андрій Володимирович, к.т.н., доцент

затверджена наказом по університету від «__» «__» р. № ____ -н

3. Строк подання студентом закінченого проекту

(роботи) «23» червня 2023 року

4. Вихідні дані до проекту (роботи):

Результати аналітичних досліджень, нормативні документи та чинне законодавство, Інтернет-джерела.

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

- зробити аналіз впливу інформаційних технологій на діяльність підприємства;

- спроектувати інформаційну підсистему матеріального забезпечення;

- реалізувати інформаційну систему, за допомогою якої можна створювати файли даних;

- реалізувати можливість перегляду, редагування, зберігання, відбір та сортування даних в інформаційній системі;

- розробка проекту інформаційної підсистеми матеріального забезпечення;

- розробка програмного додатку управління матеріально-технічним забезпеченням коледжу.

6. Дата видачі завдання:

« »

2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту(роботи)	Строк виконання етапів проекту	Примітка
1	Визначення об'єкту, предмету дослідження, формулювання мети та задач кваліфікаційної роботи, складання плану	<i>вересень-жовтень 2022 р.</i>	
2	Підбір та вивчення літературних джерел, законодавчої та нормативної бази	<i>листопад- грудень 2022 р.</i>	
3	Узагальнення теоретичного матеріалу з обраної теми дослідження та представлення чорнового варіанту першого розділу кваліфікаційної роботи	<i>січень 2023 р..</i>	
4	Збір та обробка фактичного матеріалу, узагальнення аналізу застосування досліджуваного питання в установі (переддипломна практика)	<i>лютий-березень 2023 р.</i>	
5	Оформлення теоретичної частини кваліфікаційної роботи, узагальнення аналітичної частини, захист звіту по переддипломній практиці	<i>до 10.05.2023 р.</i>	
6	Оформлення практичної частини кваліфікаційної роботи, розробка проекту, розробка програмного продукту для перевірки матеріального забезпечення	<i>до 30.05.2023 р..</i>	
7	Підготовка висновків, пропозицій та узгодження з керівником	<i>до 1.06.2023 р.</i>	
8	Підготовка кінцевого варіанту кваліфікаційної роботи	<i>02.06.2023 р.</i>	
9	Опрацювання зауважень керівника, перевірка на автентичність	<i>06.06.2023 р.</i>	
10	Перевірка з академічної доброчесності кваліфікаційної роботи	<i>09.06.2023 р.</i>	
11	Представлення кваліфікаційної роботи на кафедру, деканат	<i>12.06.2023 р.</i>	
12	Підготовка доповіді, презентації до кваліфікаційної роботи та її попередній захист	<i>23.06.2023р.</i>	
13	Захист кваліфікаційної роботи	<i>28 червня 2023р.</i>	

Студент

(підпис)

Артур ШАКОТЬКО

Керівник роботи

(підпис)

**Андрій
ТОЛБАТОВ**

Перевірку на автентичність проведено.

Надія БАРАНИК

Дипломна робота допущена до захисту

**Маргарита
ЛИШЕНКО**

АНОТАЦІЯ

Шакотько А.М. Інформаційна підсистема матеріального забезпечення інформаційної системи коледжу ВСП «ГАФК СНАУ».

Кваліфікаційна робота зі спеціальності 126 «Інформаційні системи та технології» ОП Інформаційні системи та технології, СНАУ, Суми – 2023 р. – Рукопис.

Кваліфікаційна робота складається зі вступу, 3 розділів, висновків, списку використаних джерел із 20 найменувань, додатків. Загальний обсяг роботи – 102 сторінки, у тому числі 62 сторінки основного тексту, 2 сторінки списку використаних джерел, 33 сторінки додатків.

Об'єкт роботи – інформаційна підсистема матеріального забезпечення інформаційної системи підприємства.

Предмет дослідження кваліфікаційної роботи: Розробка та оптимізація інформаційної підсистеми матеріального забезпечення для покращення управління ресурсами та ефективності інформаційної системи коледжу ВСП «ГАФК СНАУ».

Метою роботи є розробка інформаційної системи з графічним інтерфейсом, яка буде забезпечувати ефективне вирішення завдань, пов'язаних з роботою з файлами, їх збереженням та редагуванням.

У розділі 1 роботи проводиться аналіз впливу інформаційних технологій на діяльність підприємства, зокрема в контексті матеріального забезпечення. У розділі 2 роботи проводиться проектування інформаційної системи. Розділ 3 присвячений розробці та опису роботи інформаційної системи.

Результатом роботи є функціональна інформаційна підсистема матеріального забезпечення. Система дозволяє створювати та опрацьовувати бінарні файли з даними, а також забезпечує можливість перегляду, редагування, зберігання, відбору та сортування даних.

Ключові слова: інформаційна система, розробка, бінарні дані, матеріальне забезпечення.

SUMMARY

Shakotko A.M. The information subsystem of the material support of the information system of the college of VSP "HAFC SNAU".

Qualification work on specialty 126 "Information systems and technologies" OP Information systems and technologies, SNAU, Sumy - 2023 - Manuscript.

The qualification work consists of an introduction, 3 chapters, conclusions, a list of used sources of 20 names, appendices. The total volume of the work is 102 pages, including 62 pages of the main text, 2 pages of the list of used sources, 33 pages of appendices.

The object of the work is the information subsystem of the material support of the enterprise's information system.

The subject of the research of the qualification work: Development and optimization of the information subsystem of material support to improve the management of resources and the efficiency of the information system of the college SSS "HAFC SNAU".

The goal of the work is the development of an information system with a graphic interface that will provide effective solutions to tasks related to working with files, saving and editing them.

Chapter 1 of the work analyzes the impact of information technologies on the company's activities, in particular in the context of material support. In section 2 of the work, information system design is carried out. Chapter 3 is devoted to the development and description of the information system.

The result of the work is a functional information subsystem of material support. The system allows you to create and process binary files with data, and also provides the ability to view, edit, store, select and sort data.

Keywords: information system, development, binary data, material support.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1	9
АНАЛІЗ ВПЛИВУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ НА ДІЯЛЬНІСТЬ ПІДПРИЄМСТВА	9
1.1 Особливості інформаційної діяльності підприємств у перехідному (транзитивному) суспільстві	9
1.2 Основні тенденції розвитку інформаційних технологій в Україні	14
1.3 Аналіз задач управління, що реалізують інформаційний процес	19
РОЗДІЛ 2	30
ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	30
2.1 Робота з бінарними даними.....	30
2.2 Використання платформи .Net Framework при роботі з даними файлів	37
2.3 Реалізація роботи з бінарними файлами	41
РОЗДІЛ 3	48
РОЗРОБКА ТА ОПИС РОБОТИ ІНФОРМАЦІЙНОЇ СИСТЕМИ	48
3.1 Відбір даних за умовою з формуванням текстового документу	48
3.2. Створення додаткових форм та меню	55
3.3. Використання додатку	62
ВИСНОВКИ.....	66
СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	67
ДОДАТОК А ТЕХНІЧНЕ ЗАВДАННЯ	68
ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ	72
ДОДАТОК В. ЛІСТИНГ ФАЙЛІВ ПРОГРАМНОГО КОДУ	77
ДОДАТОК Г ОПУБЛІКОВАНІ ТЕЗИ	94
ДОДАТОК Д ІНСТРУКЦІЯ КОРИСТУВАЧА	99
ДОДАТОК Е ДОВІДКА ПРО ВПРОВАДЖЕННЯ.....	103

ВСТУП

В епоху інформаційного розвитку значення інформації в житті людини постійно зростає, разом зі збільшенням обсягу самої інформації. Основною темою дослідження даної роботи є інформаційний пошук. Людство вже протягом кількох десятиліть намагається вирішити проблеми, пов'язані збереженням та використанням великого обсягу даних. Одна з таких проблем стосується ефективного пошуку інформації.

Зазвичай пошук полягає у простому порівнянні тексту за ключовим словом запиту, але цей метод не є найефективнішим, оскільки не враховує змісту тексту. Розподілення інформації за темами та типами може покращити результати пошуку, але само по собі не є достатньо ефективним рішенням. Тому використання семантичної мережі та семантичних моделей може стати відповіддю на ці проблеми.

Суть використання семантичної мережі полягає в поліпшенні результатів пошуку шляхом створення семантичних зв'язків між даними. Іншими словами, використання семантичної мережі - це спроба навчити комп'ютери інтерпретувати та пов'язувати інформацію, що полегшить пошук потрібної інформації та підвищить ефективність пошукових систем.

Крім популярних операційних систем Windows, компанія також надає користувачам можливість ознайомитися з однією з найпотужніших мов програмування – C++. Програмний продукт Visual Studio не тільки є зручним у використанні, але має також значну кількість розроблених функцій та модулів. Важливим інструментом цієї програми є фреймворк .NET Framework, що дозволяє створювати програмне забезпечення з використанням графічного користувацького інтерфейсу, аналогічного тому, що використовується в операційній системі Windows.

Актуальність даної теми полягає в тому, що з розвитком комп'ютерних технологій, необхідні високо кваліфіковані працівники-програмісти, які можуть реалізовувати зв'язку між апаратурою та користувачем. Так як ОС Windows – є

найпопулярнішою системою в світі, використовуючи ПЗ для розробки інших програм, краще за все використовувати сам продукт Visual Studio, що дуже тісно пов'язаний з операційною системою. До того ж, надійшла пропозиція, щодо удосконалення підсистеми матеріального забезпечення інформаційної системи коледжу.

Проблема: використання сучасних засобів розробки для створення інформаційної системи.

Об'єкт: Інформаційна підсистема матеріального забезпечення інформаційної системи підприємства.

Предмет дослідження дипломної роботи: Розробка та оптимізація інформаційної підсистеми матеріального забезпечення для покращення управління ресурсами та ефективності інформаційної системи коледжу ВСП «ГАФК СНАУ».

Мета: Розробити інформаційну систему з графічним інтерфейсом, яка буде забезпечувати ефективне вирішення завдань, пов'язаних з роботою з файлами, їх збереженням та редагуванням. Планується розширити функціональність програми шляхом використання діалогових вікон та додаткових форм. Основним завданням є успішне вирішення поставлених завдань за допомогою створеної інформаційної системи. А також є напруцювання комплексу заходів з удосконалення управлінням матеріально – технічним забезпеченням коледжу.

Тема: інформаційна підсистема матеріального забезпечення інформаційної системи коледжу ВСП «ГАФК СНАУ».

Завдання: Розробити інформаційну систему, яка дозволить створювати та опрацьовувати бінарні і текстові файли з даними згідно з індивідуальним варіантом. У рамках дипломного проєкту необхідно виконати такі завдання:

1. Зробити аналіз впливу інформаційних технологій на діяльність підприємства.
2. Спроекувати інформаційну підсистему матеріального забезпечення.
3. Реалізувати інформаційну систему, за допомогою якої можна створювати файли даних.

4. Реалізувати можливість перегляду, редагування, зберігання, відбір та сортування даних в інформаційній системі.

Для забезпечення інформаційної частини проекту та надання інформації про автора роботи та ознайомлення з поставленими завданнями, розробити три додаткові форми: 1) форму про автора, 2) форму для виведення, редагування та форматування параметрів текстового документа 3) форму-заставку. Організувати перехід до створених форм та додаткових функцій для роботи з файлом даних за допомогою меню.

Апробація дослідження. Результати дослідження було представлено на міжнародній та університетській науково-практичних конференціях, а саме, на 27-ої міжнародній науково-практичній конференції (23-24 листопада 2021 р.) Технології ХХІ сторіччя та науково-практичній конференції викладачів, аспірантів та студентів, аспірантів Сумського НАУ(25-28 квітня 2023 р.).

РОЗДІЛ 1

АНАЛІЗ ВПЛИВУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ НА ДІЯЛЬНІСТЬ ПІДПРИЄМСТВА

1.1 Особливості інформаційної діяльності підприємств у перехідному (транзитивному) суспільстві

Поєднання науки і виробництва зумовило появу нового фактора виробництва – інформації. У перехідному суспільстві особливість інформаційної діяльності визначається процесами трансформації економічного базису на основі впливу інформації та інформаційно-комунікаційних технологій, які її породжують.

Генерація інформації відбувається за рахунок її накопичення та обробки з певних позицій. Цією властивістю володіє інформація, ставши джерелом нових даних, що ведуть до нових знань.

Залучення інформаційних ресурсів, що є результатом інтелектуальної діяльності, досягло рекордного рівня. Вона набула такого масштабу, що багато експертів говорять про «інформаційний вибух» та «інформаційну революцію».

Враховуючи праці вітчизняних і зарубіжних учених з проблем інформації, усі праці ти можеш поділити на дві групи (див. рис. 1.1) [4].

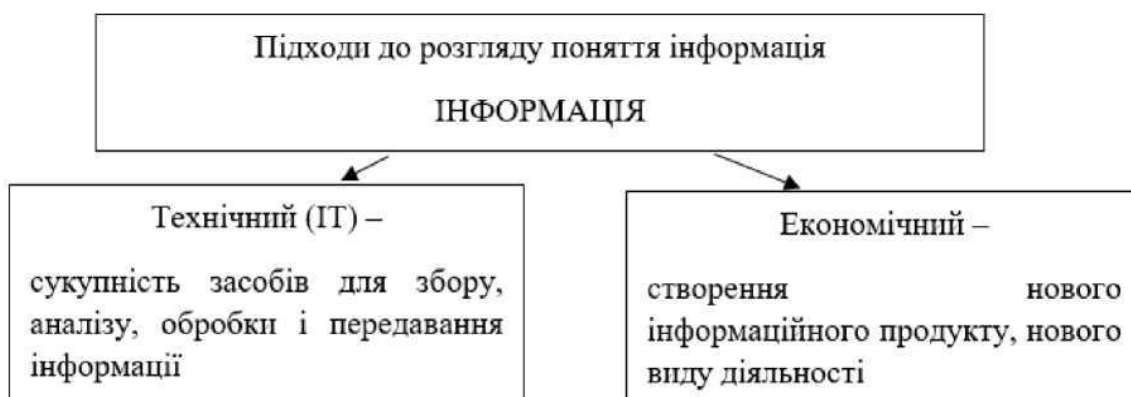


Рисунок 1.1 – Підходи до розгляду поняття «інформація»

Джерело: запропоноване автором

Розглянемо інформацію, інформаційні технології (ІТ) з технічної точки зору.

Таким чином, сучасні інформаційні технології базуються на трьох технологічних досягненнях:

1. поява нового носія для зберігання інформації на машинописних носіях;
2. розвиток засобів зв'язку, які гарантують передачу інформації за допомогою засобів зв'язку великій масі людей практично в будь-яку точку світу, без суттєвих обмежень, таких як час або відстань;
3. можливість автоматичної обробки інформації комп'ютерами за певними алгоритмами. В загальному розвитку інформаційних технологій пройшов чотири етапи, які базувалися на створенні та вдосконаленні нових засоби обробки інформації.

На початковому етапі впровадження ІТ доступ до інформаційних ресурсів був обмежений функціональними обмеженнями та високою вартістю. Основними завданнями, які вирішувалися на цьому етапі, були прискорення рутинних процесів розрахунку показників та забезпечення максимальної кількості операцій, що виконуються за одиницю машинного часу.

Другий етап еволюції ІТ характеризувався скороченням ресурсів, зокрема зменшенням вартості комп'ютерів, скороченням часу обчислень та економією людських ресурсів для обслуговування комп'ютерів.

На третьому етапі комп'ютери стали доступними для всіх людей у повсякденному житті.

Основними цілями підприємств у цей період були автоматизація робочих місць, зменшення рутинної роботи та інтелектуалізація будь-якої взаємодії з користувачами за допомогою технічних засобів. Це вимагало створення стандартних технологій для автоматизації індивідуальних знань та економії праці користувача, працівника.

Четвертим етапом стало всебічне використання інформаційних технологій у всіх сферах життя суспільства, тобто глобалізація суспільства.

Однак цей етап збігається зі становленням України як держави і є перехідним періодом. Перехідні періоди в суспільному розвитку, особливо в Україні, завжди

були складними і сповненими протиріч, часто перебувають у стані державно-правової кризи і пов'язані з критичним переосмисленням практик, які працювали в минулому [2].

Зрештою, знання про причини та умови перехідних процесів має велике практичне значення, оскільки може допомогти знайти шляхи виходу з критичних ситуацій [3].

Так, відбувся перехід від адміністративно-командної системи управління з жорстким ієрархічним контролем і штучним обмеженням інформації до ринкової економіки, де надлишок інформації робить її менш корисною на всіх рівнях управління. Як наслідок, в нашій країні існують недержавні механізми задоволення інформаційних потреб бізнес-організацій, але вони не досягли того рівня, який би сприяв сталому розвитку не лише економіки, а й суспільства в цілому.

Незважаючи на глобалізацію, під впливом ІТ змінився і той факт, що побудова перехідного суспільства в Україні є процесом перетворення вхідних ресурсів у товари, продукти та послуги.

У процесі економічної діяльності ресурси перетворюються на товари, робочі місця та послуги. Капітал, як елемент процесу виробництва програмного забезпечення, використовується для зв'язку та координації різних етапів виробничого процесу, завжди в поєднанні з необхідними компонентами, інформаційними ресурсами, пов'язаними з ними. (див. рис. 1.2)



Рисунок 1.2 – Перетворення вхідних ресурсів на товари, продукцію, послуги

Джерело: запропоноване автором

Таким чином, органи місцевого самоврядування мають найбільшу вагу в загальній структурі органів державної влади та місцевого самоврядування,

оскільки вони забезпечують можливості використання засобів електронної демократії. Державні органи, включаючи навчальні заклади, складають лише 7% від загальної кількості. (див. Рис. 1.3)

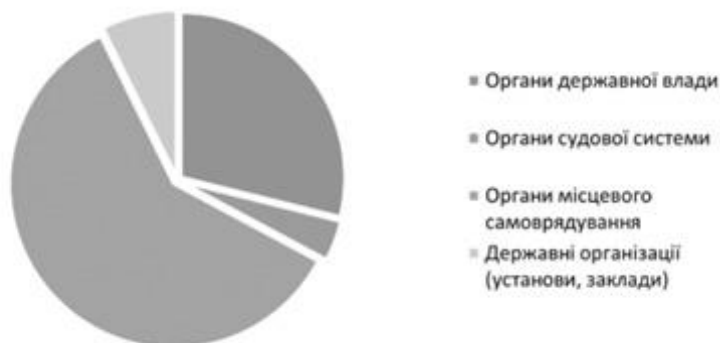


Рисунок 1.3 – Кількість органів державної влади та місцевого самоврядування, що надають можливість використання інструментів електронної демократії.

Джерело: запропоноване автором

Для ілюстрування рівня інформатизації підприємств розглянемо показники кількості підприємств, які використовують комп'ютери, та середньооблікової кількості працівників, які використовують комп'ютери, залежно від видів економічної діяльності в Україні, згідно з таблицею 1.1.

Таблиця 1.1 - Кількість підприємств, які використовували комп'ютери, та середня кількість працівників, які використовували комп'ютери.

Вид економічної діяльності	Кількість підприємств, що використовують комп'ютери				
	2018	2019	2020	2020	% від загальної кількості підприємств
Переробна промисловість	11089	11279	11469	%	
Постачання електроенергії, газу, пари та кондиційованого повітря	706	714	722	25,6	91
Водопостачання: каналізація, поводження з відходами	1147	1152	1157	11,3	93
Будівництво	4871	4971	5071	24,2	93.6
Оптова та роздрібна торгівля; ремонт автотранспортних засобів і мотоциклів	10912	10889	10866	8,3	85,4

Продовження таблиці 1.1

Транспорт, складське господарство, поштова та кур'єрська діяльність	3542	3626	3710	2,8	87,3
Тимчасове розміщування й організація харчування	1312	1286	1260	4,4	78,8
Інформація та телекомунікації	1962	1961	1960	6,1	89,9
Операції з нерухомим майном	2787	2763	2739	6,2	88,4
Професійна, наукова та технічна діяльність	2688	2729	2770	6,2	88,6
Діяльність у сфері адміністративного та допоміжного обслуговування	3052	3103	3154	7	80,2

Джерело: [21]

Тому для побудови інформаційної системи необхідно враховувати ті сектори економічної діяльності, де комп'ютерна техніка використовується найбільш інтенсивною, такі як переробна промисловість, будівництво, водопостачання, каналізація та поводження з відходами.

Однак цей показник також залежить від частки цих секторів у загальній структурі економічної діяльності в Україні. Більшість підприємств (78,8%-93,6%) вже використовують комп'ютери у своїй діяльності, але все ще існує певний відсоток підприємств, які не використовують комп'ютерні технології.

Стратегії розвитку інформаційного суспільства України до 2020 року та Стратегії сталого розвитку України до 2030 року показує, що в суспільстві перехідного періоду пріоритети зміщуються від загальної інформації до нової інформації, яка підтримує захист навколишнього середовища та сталий розвиток.

Ці стратегічні документи є основою для створення інформаційної системи, яка враховує ці пріоритети.

1.2 Основні тенденції розвитку інформаційних технологій в Україні

Вплив зовнішніх соціально-економічних процесів на економічну діяльність України є одним з факторів сучасного світу. В умовах глобалізації світова економіка у 21 столітті функціонує як система, заснована на інтенсивній конкуренції.

Індекс цифрового доступу, розроблений Міжнародним союзом електрозв'язку (МСЕ), та останні дані про перехід від технології 4G до 5G вказують на необхідність перегляду потенціалу країн з точки зору доступу до інформаційних технологій.

Раніше головною перешкодою для розвитку цифрових технологій вважалася неадекватна інфраструктура. Однак дослідження показали, що доступність послуг і рівень освіти населення є важливими факторами, взаємопов'язаними з розвитком інформаційних технологій.

У дослідженні МСЕ щодо загального потенціалу доступу країн до інформаційних технологій враховувалися питання традиційної телекомунікаційної інфраструктури, а також наявність мобільних телефонів і традиційних телефонних ліній.

Згідно даного дослідження всі країни розподілені на чотири категорії доступу до інформаційних технологій (табл.1.2) [9].

Таблиця 1.2 - Зони доступу до інформаційних технологій

Категорія зони	Країни
I. Розширений доступ	Гонконг, Фінляндія, Тайвань, Канада, США, Великобританія, Швейцарія, Сінгапур, Японія, Люксембург, Австрія, Бельгія, Нова Зеландія, Італія, Франція, Словенія, Ізраїль
II. Широкий доступ	Ірландія, Іспанія, Чехія, Греція, Португалія, ОАІ, Макао, Угорщина, Польща, Словачки
III. Середній доступ	Ліван, Таїланд, Румунія, Туреччина, Македонія, Панама, Боснія, Колумбія, Йорданія, Китай, Україна, Росія, Молдавія

Продовження таблиці 1.2

IV. Низький доступ	Білорусь, Зімбабве, Гондурас, Сирія, Пакистан, Азербайджан, Таджикистан, Кенія, Нікарагуа
--------------------	---

Джерело : [9]

Країни, які мають широкий доступ до інформаційних технологій, використовують їх як каталізатор для вирішення економічних, соціальних і політичних проблем. Держави також впроваджують політику, спрямовану на досягнення і підтримку високого рівня доступу до інформаційних технологій.

Оскільки населення планети продовжує зростати, збільшується і кількість користувачів Інтернету. У 2020 році мережею користувалося 49% населення планети, а в 2021 році ця цифра зросла до 61%. Найвищий рівень користувачів Інтернету спостерігається в Європі, де він становить 85% загального населення, а в Північній Америці - 84%. Найнижчий рівень спостерігається в Південно-Східній Азії та Океанії, що має потенціал для майбутнього розвитку [9].

Оцінка доступу до цифрових технологій базується на кількох показниках, рекомендованих Міжнародним Союзом Електрозв'язку (ITU). Серед найважливіших з них:

- Кількість підприємств, що надають послуги зв'язку.
- Кількість абонентських телефонних ліній на 100 жителів.
- Кількість мобільних телефонів на 100 жителів.
- Кількість користувачів Інтернету на 100 жителів.
- Кількість підприємств, які мають доступ до Інтернету.
- Розподіл доходів від надання послуг доступу до Інтернету.
- Кількість підприємств зі спеціалістами у сфері інформаційно-комунікаційних технологій.
- Ступінь розвитку електронної комерції в країні.

Щодо доступу до Інтернету в Україні, у 2020 році 88% підприємств мали доступ до Інтернету за всіма видами діяльності, але у 2021 році цей показник знизився до 86,4%. Щодо кількості користувачів Інтернету на 100

домогосподарств, за оцінками аналітиків на 2024 рік цей показник становив 82 користувачі на 100 домогосподарств.

Таблиця 1.3 - Напрями використання мережі Інтернет підприємствами України

Напрямок	Кількість підприємств		Частка в загальній структурі, %
	2020	2021	
Використання електронної пошти	42733	43278	18,2
Здійснення телефонних дзвінків за допомогою Інтернет-зв'язку або відеоконференцій	13763	14450	6,1
Отримання інформації про товари та послуги	38468	39066	15,0
Користування миттєвим обміном повідомленнями та електронною дошкою оголошень	21167	21654	16,4
Отримання інформації від органів державної влади	35034	35671	9,1
Здійснення операцій з органами державної влади (за винятком отримання інформації)	22413	23361	9,8
Здійснення банківських операцій	42070	42754	17,9
Доступ до інших фінансових послуг	17612	18147	7,6

Джерело: [11-14]

Ділове спілкування через електронну пошту (18,2%), банківські платежі (17,9%), обмін миттєвими повідомленнями та використання електронних дошок оголошень (16,4%), а також отримання інформації про товари та послуги (15%) є основними способами, за допомогою яких підприємства використовують Інтернет.

Серед інших інтернет-послуг, якими підприємства користуються рідше, - телефонні дзвінки через Інтернет/VoIP або відеоконференції, отримання інформації від органів державної влади та ведення бізнесу з органами державної влади (окрім отримання інформації).

Згідно з аналізом вітчизняних та зарубіжних економістів, встановлено, що для досягнення одиничного економічного зростання країни необхідно забезпечити тричасткове зростання телекомунікаційної галузі [1].

Тому однією з найважливіших проблем, яка стоїть перед нашою країною, є інформатизація суспільства, яка описана у "Концепції стійкого розвитку" [2].

Одним із головних пріоритетів концепції є досягнення оптимального балансу між економічним, соціальним і екологічним розвитком. Інформаційні технології та

Інтернет є ключовими елементами, які створюють взаємозв'язки у країні.

Ці процеси, на відміну від загального уявлення, не полягають у простому поширенні Інтернет-технологій, а навпаки - Інтернет-технології з'явилися у відповідь на потребу в них. В електронній комерції існують три основні класи: B2B (від бізнесу до бізнесу), B2C (від бізнесу до споживача) і B2G (від бізнесу до уряду) [3].

Аналіз показав, що найбільш розвиненим є напрям B2B і B2C, який характеризується зростанням Інтернет-торгівлі як для бізнесу, так і для споживачів.

Таблиця 1.4 - Обсяги електронної торгівлі у світі, млрд. дол. США

Форма е-торгівлі	2017 р.	2018 р.	2019р.	2020р.	2021 р.
B2B	604	1138	2061	3694	6335
B2C	53	96	169	284	452

Джерело: [21]

Отже, електронна комерція зростає. Тому що онлайн-покупки - це наступний крок у впровадженні принципів управління електронною комерцією. За прогнозами експертів, до кінця 2021 року онлайн-продажі досягнуть 2 трильйонів доларів США.

Основними конкурентами на ринку є Китай і Північна Америка (рис.1.4).

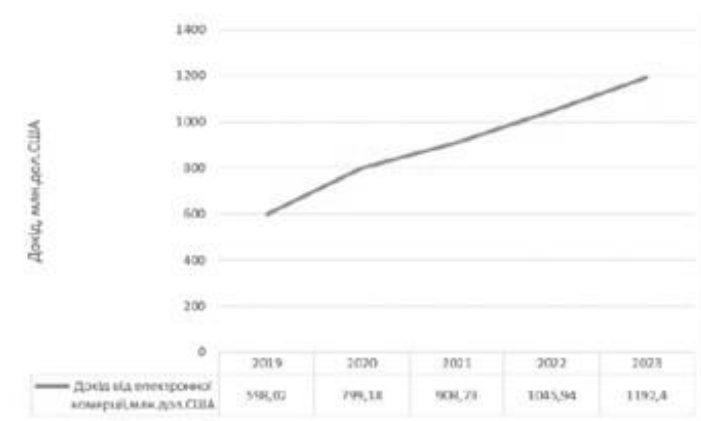


Рисунок 1.4. – Дохід від електронної комерції в США, млн.дол.

Джерело: [9]

Так, пандемія коронавірусу позитивно вплинула на ринок електронної

комерції, оскільки доходи від такої діяльності в США зросли на 13,2%. Наступного року очікується, що доходи від цього виду діяльності зростуть на 15,5%, досягнувши 1 459,4 млрд. дол.

Розглянемо товари, що мають найбільшу популярність в мережі (рис.1.5)



Рисунок 1.5 - Структура продажів товарів в сфері е-комерція

Джерело: [9]

Як можна зазначити, основна частина товарів, які продаються через Інтернет, включає одяг і аксесуари, товари для дому. Асортимент продукції є дуже різноманітним.

З урахуванням сучасних тенденцій давайте розглянемо переваги і недоліки електронної комерції наведено в таблиці 1.5.

Таблиця 1.5 - Переваги і недоліки електронної комерції

Переваги	Недоліки
Нижчі ціни у порівнянні з традиційними магазинами	Завжди потрібен доступ до мережі Інтернет
Купівля товарів 24/7 без обмежень	Відсутність повного законодавчого регулювання
Більше інформації про товар	Високий рівень шахрайства
Зручність оплати	Складність з перевіркою обсягів замовлень і сплатою податків

Джерело : [16]

Інформаційні системи є ключовими складовими цього процесу і виконують важливі функції: обробка, збереження і передача даних, забезпечення безпеки та управління ресурсами. Тому, разом з перевагами і недоліками електронної комерції, важливо розглянути вплив інформаційних систем у цьому контексті.

1.3 Аналіз задач управління, що реалізують інформаційний процес

Сучасний світ визнає інформацію як стратегічний національний ресурс, що має значний вплив на розвиток та ефективність систем управління. Застосування сучасних інформаційних технологій забезпечує прямий вплив на суспільство.

Використання сучасних технологій інформаційного управління означає здійснення різноманітних інформаційних операцій для вирішення завдань як на рівні окремого підприємства, так і на рівні держави загалом.

Детальне дослідження процесів комп'ютеризації управління в різних сферах свідчить про те, що органи управління на підприємствах відчують недостатнє інформаційне забезпечення своєї діяльності через наступні причини:

1. Відсутність аналітичного відділу та недостатнє технічне оснащення, зокрема засобів зв'язку.

2. Зростання вимог до якості ухвалених рішень, що залежить від професійної підготовки аналітиків та впровадження сучасних інформаційних технологій у процесу прийняття рішень.

3. Збільшення обсягу та потоків даних, як в межах країни, так і за її межами, настільки значне, що без структурування та використання інформаційних систем процес обміну інформацією не може бути ефективним.

Інформаційні системи в діяльності українських підприємств мають законодавчу підтримку, яка відображена в наступних законодавчих актах: Закон України "Про Національну програму інформатизації", Указ Президента України "Про вдосконалення інформаційно-аналітичного забезпечення діяльності Президента України та органів державної влади", Постанова Кабінету Міністрів України "Про затвердження Завдання Національної програми інформатизації" та "Про затвердження Порядку формування та реалізації галузевої програми та проекту інформатизації".

Інформація передається, зберігається, обробляється і об'єднується в системі управління, утворюючи значущі групи. Ця асоціація має назву "документація", яка є джерелом інформації, необхідної для аналізу та моніторингу даних. З цього

впливає поняття документообігу. Документообіг - це процес переміщення документів всередині системи управління, від місця їх створення до використання у процесі управління – від джерела їх формування до використання в управлінні [5].

Одним із основних елементів інформаційних систем є документообіг, який забезпечує передачу, обробку та зберігання інформації у системі управління. Документообіг є процесом, що включає переміщення документів від їх початкового джерела до використання в управлінні. В рамках документообігу інформація групується, асоціюється і структурується для подальшого аналізу та моніторингу даних.

Українські підприємства мають законодавчу базу для реалізації інформаційних систем. Національна програма інформатизації, встановлена Законом України, визначає стратегічні пріоритети в галузі інформаційних технологій. Крім того, Указ Президента України та Постанова Кабінету Міністрів України стосуються питань покращення інформаційно-аналітичного забезпечення діяльності державних органів.

У сучасному світі інформація є важливим ресурсом, що впливає на розвиток та ефективність систем управління. Використання сучасних інформаційних технологій у процесі управління дозволяє здійснювати різноманітні інформаційні операції для вирішення завдань як на рівні окремого підприємства, так і на рівні держави в цілому.

Аналізується процес комп'ютеризації управління в різних галузях, виявлено недостатнє інформаційне забезпечення органів управління підприємств, що пов'язане з відсутністю аналітичних служб, недостатнім технічним оснащенням і обмеженими засобами зв'язку. Підвищення вимог до якості прийнятих рішень також ставить вимоги до професійної підготовки аналітиків та впровадження сучасних інформаційних технологій у процес прийняття рішень.

Зростання обсягу та потоків даних як в межах країни, так і за її межами, створює потребу в ефективному обміні інформацією. Інформаційні системи, здатні структурувати та систематизувати дані за допомогою системи договорів, стають

невід'ємною частиною цього процесу.

Система управління підприємством отримує документи з різних джерел зовнішнього середовища, таких як клієнти, постачальники, споживачі, вищі контролюючі організації та банки. Крім того, на підприємстві створюються внутрішні документи. Усі ці документи можна розділити на дві категорії: загального призначення (управлінські) та специфічні для діяльності об'єкта, які можуть бути вхідними, вихідними та внутрішніми.

Створення системи управління і документообігу на кожному підприємстві набуває важливості. Це дозволяє підготувати, зберігати та обробляти документи, відстежувати їх прогрес, керувати архівами та змінювати формати. Документи виступають як основний засіб представлення законно придбаної інформації про об'єкт управління і становлять основну частину інформаційних потоків у системі управління.

Кількість, якість і доступність інформаційного ресурсу мають вирішальне значення для розвитку підприємства та його конкурентного статусу. Інформаційний ресурс є капіталом підприємства, тому формування, накопичення та комплексне використання інформаційного ресурсу відносяться до завдань особливої важливості.

Аналіз розвитку, впровадження та функціонування інформаційних систем на підприємствах в Україні свідчить про значний прогрес за останні роки у використанні інформаційних технологій. Практично всі підприємства забезпечують інформаційне забезпечення для поточної роботи та використання персональних комп'ютерів з встановленими типовими програмними продуктами. Більшість підприємств також використовують локальні обчислювальні мережі.

Зараз, в умовах пандемії коронавірусу, виникає особлива необхідність у створенні інформаційно-аналітичної бази для прийняття управлінських рішень на підприємствах. Це стає єдиним вихідним шляхом зі складної ситуації. Тому, на підприємствах дедалі більш актуальною стає реалізація системи інформаційного моніторингу.

Завдання формування, накопичення та комплексного використання

інформаційного ресурсу є особливо важливими, оскільки сьогоднішній рівень розвитку підприємства і його конкурентоспроможність значно залежать від кількості, якості та доступності цього ресурсу.

Робота, проведена в сфері розробки, впровадження та функціонування інформаційних систем на підприємствах в Україні, свідчить про значний прогрес, досягнутий за останні роки в застосуванні інформаційних технологій. Багато підприємств забезпечують себе необхідним інформаційним забезпеченням для поточної роботи та використання комп'ютерів з встановленими стандартними програмними продуктами.

Проте, у зв'язку зі змінами в економічному середовищі та вимогами часу, виникає потреба в подальшому розвитку і вдосконаленні інформаційних систем на підприємствах, зокрема у створенні більш прогресивних систем інформаційного моніторингу, які забезпечать більш ефективне управління та прийняття стратегічних рішень.

Важливо розробляти і впроваджувати інформаційні системи, які забезпечать збір, аналіз і збереження великої кількості даних, а також забезпечать швидкий та зручний доступ до них.

Подальше розвиток інформаційних систем на підприємствах передбачає використання новітніх технологій, таких як штучний інтелект, аналітика даних, хмарні рішення та інтернет речей. Впровадження таких інноваційних рішень дозволить підприємствам забезпечити високу продуктивність, оптимізувати бізнес-процеси та покращити прийняття управлінських рішень.

Важливим аспектом розвитку інформаційних систем на підприємствах є також забезпечення кібербезпеки. З огляду на зростаючу кількість кібератак та загроз, необхідно розробляти та впроваджувати заходи з захисту інформації та персональних даних підприємства.

Усі ці заходи спрямовані на створення сучасних інформаційних систем, які допоможуть підприємствам ефективно працювати, приймати обґрунтовані рішення та відповідати викликам сучасного бізнес-середовища. Це дозволить підприємствам підтримувати свою конкурентоспроможність і досягати успіху на

ринку.

Давайте розглянемо використання інформаційного моніторингу у різних галузях. Наприклад, у сфері телекомунікацій в Україні вже у 2003 році було зареєстровано проект Закону "Про моніторинг телекомунікацій". Цей законопроект був розроблений Службою безпеки України згідно з положеннями Варшавської конференції щодо спільної боротьби з тероризмом, затвердженими постановою Кабінету Міністрів України.

Метою цього законопроекту є створення механізму, який дозволить правоохоронним органам отримувати інформацію з телекомунікаційних мереж з метою запобігання або розкриття злочинів, включаючи кіберзлочини та тероризм.

Він також передбачає співпрацю з правоохоронними органами інших країн для моніторингу телекомунікацій у боротьбі зі злочинністю. Основна мета прийняття цього законопроекту - забезпечення ефективного здобуття інформації та взаємодії між правоохоронними органами для забезпечення безпеки та протидії злочинності.

Однак запропоноване законодавство є обмеженим, зосереджується лише на боротьбі з тероризмом і не враховує моніторинг інформації про економічні аспекти телекомунікацій.

Моніторинг також широко використовується в екологічній сфері. Відповідно до вимог Закону України "Про охорону навколишнього природного середовища", Міністерство охорони навколишнього природного середовища з вересня 1992 року створило Державну інформаційну систему моніторингу екологічної, ядерної та радіаційної безпеки (ДІСМ), а з серпня 1996 року - екологічну підсистему Державної інформаційно-аналітичної системи з питань надзвичайних ситуацій.

У 1996 році були розроблені технічні умови на компоненти першої черги КЕП, затверджені основні нормативні документи на систему, виготовлені зразки обладнання для моніторингу концентрацій шкідливих речовин в атмосферному повітрі, а також стаціонарні зразки. Також створено мобільну лабораторію для моніторингу якості повітря. Ці прилади можуть використовуватися як у загальнодержавній, так і у відомчій системах моніторингу.

Ще одним важливим аспектом організації інформаційного моніторингу є фінансово-економічне забезпечення. Ця ситуація стає особливо гострою в умовах недостатнього фінансування з боку держави для підтримки розвідувальної діяльності. Це може призвести до комерціалізації джерел інформації. У таких випадках важливо визначити, які національні джерела інформації повинні надаватися безкоштовно, а які платно і за якою ціною.

Для проведення ефективної фінансово-економічної політики щодо створення та використання національних інформаційних ресурсів необхідно розробити правила фінансування та звітності про створення та використання інформаційних ресурсів на основі положень Закону України "Про інформацію" та Національної програми інформатизації.

Поняття функції спостереження спочатку (у 80-х роках 20 століття) розглядалося як інформаційна система (ІС), але з розвитком комп'ютерних технологій воно набуло самостійного визначення і поступово перейшло з технічної в економічну сферу. Фактично, поняття інформаційного "моніторингу" з економічної точки зору нерозривно пов'язане з поняттям економічної інформації та її матеріального представлення.

Варто розрізняти два аспекти моніторингу: змістовний та технічний. Змістовний аспект визначає коло управлінських проблем, які вирішуються за допомогою моніторингу. Він відображає структуру об'єкта управління та сфери діяльності його різних підрозділів. Технічний аспект інформаційного моніторингу розглядається як один з управлінських процесів, пов'язаний з реалізацією сукупності інформаційних процедур.

Інформаційна процедура - це сукупність однорідних операцій, які впливають на інформацію. Сюди входять процеси збору, обробки, аналізу, зберігання, передачі та використання інформації.

Характеристики інформаційних процедур можуть бути наведені в таблиці 1.6.

Таблиця 1.6 - Характеристика інформаційних процедур

Інформаційна процедура	Характеристика	Супутні інформаційні процедури
Збирання даних, реєстрація	Мета — одержати точне, своєчасне, вірогідне та повне відображення явищ економічного життя (виробничо-господарської діяльності). Дані збираються всередині об'єкта і надходять із зовнішнього середовища	Кодування одиниць інформації (присвоєння кодових позначок)
Передача даних	Два варіанти організації передачі даних: - фізичне переміщення носіїв (кур'єром, транспортом, поштою); - дистанційна передача по лініях зв'язку інформації носіїв у вигляді сигналів	Захист інформації найпотужнішими механізмами захисту є криптографічні методи: шифрування дешифрування)
Збереження інформації	Інформаційна процедура пов'язана з багаторазовістю, тривалістю використання умовно-сталої інформації в обробленні, необхідністю накопичення інформації у зв'язку з розривом у часі її оброблення	Пошук одиниць інформації. Захист від несанкціонованого доступу. Актуалізація даних - підтримка значень одиниць інформації на заданому рівні
Оброблення інформації	Сукупність арифметичних і логічних операцій для здобуття результативної інформації (інформації для управління)	Пошук одиниць інформації. Розмноження результативної інформації
Передача результативної інформації	Два варіанти організації передачі: - передача інформації для: використання в управлінні, збереження; - передача по каналах зв'язку: а) для використання в управлінні: б) для збереження,	Захист інформації. Збереження інформації
Споживання результативної інформації	Використання результатів оброблення у вигляді розрахованих значень техніко-економічних показників (ТЕП) для визначення факторів, що впливають на виробництво, і виявлення резервів розвитку	Актуалізація даних

Джерело: [11]

Враховуючи інформаційні процедури, такі як збір даних, їх реєстрація, передача даних, зберігання та обробка інформації, споживання ефективної інформації, а також основні характеристики інформації, такі як ясність, достовірність та актуальність, можна визначити основні напрямки використання ефективної інформації на підприємстві (див. Рисунок 1.6).

З цих вихідних даних можна стверджувати, що вся сукупність інформаційних

процедур утворює інформаційний процес, який є технологічною основою управлінської діяльності [11].

Процес управління інформацією складається з двох основних частин: інформаційного процесу та процесу прийняття управлінських рішень. Інформаційний процес включає збір, обробку, аналіз та поширення інформації, а процес прийняття управлінських рішень передбачає оцінку інформації, формування альтернативних варіантів і вибір оптимального рішення на основі отриманої інформації.



Рисунок 1.6 – Напрями використання результативної інформації

Джерело: запропоноване автором

Так, інформаційний процес можна розглядати як процес створення інформаційного продукту, подібний до виробництва, який має на меті отримання цілісної сукупності даних для обробки та прийняття управлінських рішень.

Інформаційний процес включає комунікаційний процес, який має наступні функції на підприємстві (див. таблицю 1.7):

1. Обробка текстових документів: ця функція включає аналіз та обробку документів, зокрема текстових матеріалів, що надходять на підприємство.

2. Пошук необхідних відомостей: комунікаційний процес включає пошук необхідних даних, інформації та відомостей у довідкових фондах та архівах підприємства.

3. Формування документів: ця функція передбачає створення і формування документів, таких як відомості, таблиці та інші документи, які необхідні для подальшого використання.

4. Подання інформації: комунікаційний процес включає своєчасне подання інформації керівництву, органам та органам управління підприємством.

5. Уточнення та модернізація завдань управління: ця функція передбачає узгодження, уточнення та модернізацію завдань управління на основі отриманої інформації.

6. Прийняття рішень в умовах невизначеності або браку інформації: комунікаційний процес сприяє прийняттю рішень в умовах невизначеності або обмеженості наявної інформації шляхом усунення браку

7. Контроль якості інформації: ця функція передбачає перевірку та контроль якості інформації, що використовується в процесі управління. Вона включає перевірку достовірності, актуальності та повноти інформації.

8. Аналіз та оцінка інформації: комунікаційний процес включає аналіз та оцінку отриманої інформації з метою виявлення тенденцій, проблем або можливостей для подальшого управління.

9. Забезпечення конфіденційності інформації: ця функція передбачає захист інформації від несанкціонованого доступу та збереження конфіденційності даних.

10. Розповсюдження інформації: комунікаційний процес забезпечує розповсюдження інформації серед відповідних структур та підрозділів підприємства.

Ці функції інформаційного процесу на підприємстві допомагають забезпечити збір, обробку, передачу та використання інформації для прийняття управлінських рішень [11-14].

Таблиця 1.7 - Функції інформації на підприємстві

Найменування виду робіт	Перелік функцій
Текстове оброблення документів	Підготовка документів (введення інформації та її коригування); форматизація документів; автоматизоване редагування документів і текстів; включення в текст стандартних фраз; композиція документів; друк; тиражування документів
Інформаційний пошук	Перегляд каталогів документів і файлів; пошук за ключами та іменами документів і файлів; пошук за фразами та контекстними виразами; формування результатів пошуку і складання вихідних документів та звітів
Ведення БД та архівів користувача	Визначення структур даних та їх взаємозв'язку; створення БД; введення, контроль та індексування даних; реструктуризація даних; підтримка цілісності та відновлення даних; забезпечення конфіденційності даних, що містяться в архіві користувача.
Оброблення інформації	Формування запитів на оброблення; формування форм звітів і документів; створення бібліотек звітів; статистичний аналіз на підставі інформації БД; вибіркове оброблення даних
Електронна пошта	Забезпечення комунікацій між користувачами; введення, ідентифікація повідомлень, редагування і компонування повідомлень, передача повідомлень, адресату збереження повідомлень у «поштових скриньках»

Джерело: [12]

Ми звернули увагу на важливі аспекти інформатизації управлінської діяльності на підприємстві. Інформатизація стала необхідним ресурсом для досягнення ефективного управління і прийняття правильних рішень. Вона допомагає впорядкувати інформацію, забезпечити її достовірність та своєчасність.

Сучасний розвиток інформаційних технологій змушує підприємства зосередитися не лише на матеріальних ресурсах, але й на інтелектуальній складовій. Значення інформаційного капіталу, такого як технологічна та управлінська інформація, інтелектуальна власність, інновації, рекламні продукти та імідж підприємства стає все більш вагомим.

Одним зі складових успішного управління інформаційними ресурсами є облік інтелектуальної власності підприємства. Це охоплює виявлення та управління накопиченою технологічною та управлінською інформацією, інноваціями, винаходами, а також рекламними продуктами та іміджем.

Врахування цих аспектів допомагає підприємству оцінити свій потенціал та використати його в конкурентному середовищі.

Отже, інформатизація є ключовим елементом розвитку сучасного підприємства і дозволяє ефективно управляти інформаційними ресурсами, забезпечуючи прийняття обґрунтованих та успішних управлінських рішень.

Інформатизація підприємства передбачає комплексний процес, що складається з трьох основних частин: комп'ютеризації, медіатизації та інтелектуалізації. Комп'ютеризація спрямована на розширення, вдосконалення та обробку інформації, що надходить від джерела її формування.

Медіатизація передбачає розвиток і вдосконалення засобів збору, зберігання та розповсюдження інформації. Інтелектуалізація спрямована на підвищення інтелектуального потенціалу підприємства, розвиток знань і вмінь працівників у сприйманні та генерації інформації з урахуванням можливостей штучного інтелекту.

Ці три процеси взаємодіють між собою і сприяють створенню ефективної інформаційної системи на підприємстві. Інформатизація дозволяє не лише оптимізувати роботу з інформацією, але і підвищує конкурентоспроможність підприємства шляхом забезпечення доступу до достовірної та своєчасної інформації, підвищення продуктивності праці та прийняття обґрунтованих управлінських рішень.

Отже, розвиток і впровадження інформаційних систем на підприємстві є невід'ємною складовою успішної інформатизації, що сприяє підвищенню ефективності управління та забезпеченню конкурентних переваг на ринку.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Робота з бінарними даними

В сучасному світі обробка бінарних даних стала необхідною складовою різноманітних інформаційних систем. Різноманітні формати даних, такі як зображення, звукові файли, відео та багато інших, використовуються в широкому спектрі сфер, включаючи медіа, комп'ютерні ігри, медичну діагностику, фінансові послуги та багато інших.

У цьому розділі дипломної роботи ми розглянемо використання платформи .Net Framework при роботі з бінарними даними файлів. .Net Framework - це потужний інструментарій розробки програмного забезпечення, який надає широкий спектр можливостей для роботи з даними. Використання .Net Framework дозволяє зручно та ефективно опрацьовувати бінарні файли, забезпечуючи швидкий доступ до даних, їх зчитування, запис та маніпуляції.

Основною метою нашого дослідження є вивчення різних аспектів використання платформи .Net Framework при роботі з бінарними даними файлів. Ми будемо розглядати різноманітні типи бінарних даних та їх формати, і досліджувати методи їх опрацювання та аналізу з використанням інструментів .Net Framework.

Під час нашого дослідження ми зосередимось на таких аспектах, як зчитування та запис бінарних даних, маніпулювання вмістом файлів, валідація та перевірка цілісності даних, компресія та розпакування, шифрування та розшифрування, а також інтеграція з базами даних для збереження та обробки бінарних даних.

У сучасному світі велика кількість даних зберігається у бінарних форматах, і важливо мати інструменти, які дозволяють нам ефективно опрацьовувати ці дані. Використання платформи .Net Framework надає нам такі можливості, забезпечуючи

набір готових класів, бібліотек та інструментів, що спрощують роботу з бінарними даними.

Основною метою нашого дослідження є розробка та проектування інформаційної системи, яка базується на платформі .Net Framework та забезпечує ефективну роботу з бінарними даними файлів. Ми плануємо дослідити різні аспекти роботи з бінарними даними, включаючи їх зчитування, запис, маніпулювання та аналіз.

Під час нашої роботи ми зосередимось на вивченні різних форматів бінарних даних, таких як зображення, звукові файли, відео та інші, і дослідженні методів їх обробки. Ми також розглянемо технології компресії та розпакування бінарних даних, їх шифрування та розшифрування та керування бінарними даними.

Важливим аспектом нашого дослідження буде також вивчення можливостей оптимізації роботи з бінарними даними, щоб забезпечити швидкий доступ до інформації та ефективне використання ресурсів системи.

Файл - це іменована область пам'яті, призначена для тривалого зберігання інформації на зовнішньому носії. Файли мають імена і організовані в ієрархічну деревоподібну структуру, що складається з каталогів (папок) і простих файлів.

Часто використовувана метафора: якщо уявити файли як книги (тільки для читання) і зошити (для читання і запису) в бібліотеці, то відкриття файлу - це те саме, що вибір книги або зошита за назвою і відкриття обкладинки (першої сторінки).

Після відкриття ви можете читати книгу, додавати елементи, видаляти, редагувати і переміщатися по ній. Ви можете порівняти сторінку з файловим блоком, а книжкову полицю з каталогом.

Щоб отримати доступ до файлових даних з вашої програми, вам потрібно визначити функцію, яка відкриває або створює файл, і таким чином встановити зв'язок між ім'ям файлу і конкретною файловою змінною у вашій програмі. Доступ до файлових даних здійснюється з позиції читання-запису, яка автоматично просувається під час операції читання-запису. Однак існують функції для вільної зміни цієї позиції.

C++ надає засоби для роботи з двома типами файлів: текстовими та двійковими файлами.

Текстові файли призначені для зберігання тексту, тобто послідовності рядків змінної довжини. Кожен рядок закінчується керуючою послідовністю '\n', а розділювачами для слів і чисел у рядку є пробіли і табуляція.

Оскільки вся інформація в текстових файлах є символьною, програмна обробка таких файлів складається з читання рядків, видалення слів з рядків і, при необхідності, перетворення числових рядків в числові значення за допомогою відповідних функцій перетворення. Текстові файли можна створювати і редагувати в програмах, а також у текстових редакторах, таких як Word, WordPad і Блокнот.

Двійкові файли зберігають дані в тому ж форматі, що й оголошені, і їхній вигляд ідентичний тому, який вони мають у пам'яті комп'ютера. Як наслідок, вони використовують набагато менше пам'яті, ніж текстові файли з аналогічною інформацією, оскільки немає необхідності використовувати роздільники, такі як пробіли або керуючі послідовності.

Вони також не потребують можливості перетворення числових даних. Однак кожна обробка даних у бінарному файлі можлива лише за наявності програми, яка точно знає, що саме і в якому порядку зберігається в цьому файлі.

У цій статті ми розглянемо основні способи створення та обробки текстових файлів за допомогою спеціального простору імен System::IO в .NET Framework, але слід зазначити, що існують і інші підходи до обробки файлів [18].

Деякі класи простору імен System::IO для роботи з файлами

Простір імен System::IO платформи .NET Framework містить декілька класів для виконання операцій над потоками даних, файлами та каталогами (папками, директоріями), таких як читання та запис даних, пошук (визначення та зміна поточної позиції в потоці), архівування файлів та каталогів.

Існує декілька класів для виконання цих операцій. Наприклад, клас Stream Writer у цьому просторі імен має сучасні, потужні і дуже корисні інструменти для запису даних у файли у вигляді потоку даних.

Таблиця 2.1 – Класи та їх опис

Клас	Опис
File	Надає методи створення, копіювання, видалення, переміщення та відкриття файлів.
File Info	Надає приклади методів створення, копіювання, видалення, переміщення та відкриття файлів.
Directory	Надає статистичні методи для створення та переміщення каталогів.
Directory Info	Надає зразки методів створення та переміщення каталогів.
Path	Виконує операції над інформацією про шлях до файлу або каталогу (незмінний рядок (тип String), що вказує на розташування (адресу каталогу) файлу у файловій системі).
Stream Reader	Надає засоби для читання послідовності символів (байт) з потоку із заданим кодуванням.
Stream Writer	Надає засоби для запису масивів символів (байт) у потік із заданим кодуванням.
Binary Reader	Надає засоби для читання простих типів даних (двійкових значень).
Binary Writer	Надає засоби для запису простих типів даних (двійкових значень).

Джерело: [7]

Розглянемо найпоширеніші методи цих класів, які дозволять створити файл, записати у нього дані, прочитати їх та ін.

Таблиця 2.2 – Варіанти дії для вирішення відповідних задач

Дії	Методи
Відкрити файл з вказаним режимом та доступом	File::Open File::OpenRead-відкриває файл у режимі читання File::OpenWrite-відкриває файл у режимі запису
Закриття файлу(потіку)	Close
Запис у файл	Write – записати значення аргументу у потік Write Line – записати рядок з символом '\n' (Enter)
Додавання до текстового файлу	Метод File Info::Append Text відкриває існуючий файл і додає до нього текст у кодуванні UTF-8 (створюється, якщо файл не існує).
Читання з файлу	Read – зчитати з потоку значення аргументів, записаних у дужках Read Line – зчитати рядок і символ '\n' як ознаку кінця рядка File::ReadAllText – відкрити текстовий файл, зчитати всі рядки файла в один зазначений рядок і закрити файл

Продовження таблиці 2.2

Створення текстового файлу	File::CreateText або File::Create File::CreateText або FileIn- fo::Create
Перемістити позицію файлу на offset байтів відносно позиції Seek Origin: Seek Origin:Begin або Seek Origin::End	BaseStream::Seek(offset, SeekOrigin) – для текстових файлів, а для бінарних файлів: BinaryWriter::Seek(offset, SeekOrigin) Наприклад: // на початок файла f->BaseStream->Seek(0, SeekOrigin::Begin); // у кінець файла f->BaseStream->Seek(0, SeekOrigin::End);
Повернути наступний доступний символ, але не використовувати його	StreamReader::Peek З цим методом можна організувати цикл для почергового опрацювання вмісту файлу: while (fr->Peek() > -1) ...
Перейменувати чи перемістити файл	File::Move або FileInfo::MoveTo
Видалити файл	File::Delete або FileInfo::Delete
Копіювати файл	File::Copy або FileInfo::CopyTo
Визначити наявність файлу	File::Exists
Отримати дані про розмір файлу	властивість FileInfo::Length
Отримати розширення імені файлу	Path::GetExtension
Отримати повний шлях до файлу	Path::GetFullPath
Отримати ім'я й розширення файлу	Path::GetFileName

Джерело: [7]

Іншими словами, для кожної дії при роботі з файлом існує кілька способів її виконання. Розглянемо деякі з цих способів.

Створити текстовий файл з іменем MyTest.rtf у поточному каталозі можна наступною командою:

```
StreamWriter^ = File::CreateText("MyTest.rtf");
```

Також можна виконати наступну команду:

```
TextWriter^ ft = gcnew StreamWriter(File::OpenWrite("MyTest.rtf"));
```

Якщо файл, що створюється, має бути розміщено у каталозі, відмінному від поточного, необхідно вказати повний шлях до файлу:

```
StreamWriter^ = File::CreateText("c:ⒶCREPEMyTest.rtf");
```

Виконайте наступну команду для створення бінарного файлу з іменем Test.dat у поточному каталозі:

```
BinaryWriter^fb=gcnew
```

```
BinaryWriter^fb=gcnew
```

```
BinaryWriter(File::Open("Test.dat", FileMode::OpenOrCreate));
```

Або двома командами

```
FileStream^ data = gcnew FileStream("Test.dat", FileMode::OpenOrCreate);
```

```
BinaryWriter^ fb = gcnew BinaryWriter(data);
```

У цьому випадку режим відкриття `OpenOrCreate` не перезаписує існуючий файл (якщо такий є), а лише відкриває його.

Якщо файл, який створюється, знаходиться в каталозі, відмінному від поточного, необхідно вказати повний шлях до нього.

Метод `WriteLine` може бути використаний для запису рядка `s` у файл:

```
String^ s = "Текст, який потрібно записати у файл окремим абзацом".
```

```
ft->WriteLine(s);
```

Щоб додати дані в кінець файлу, скористайтеся наступною командою.

```
File::AppendAllText("MyTest.rtf", "Текст для додавання у файл");
```

У цьому випадку буде створено файл з таким ім'ям, якщо він ще не існує. Іншими словами, метод `AppendAllText` поєднує в собі функції одразу декількох методів: створення файлу, його відкриття та додавання даних.

Щоб прочитати рядки текстового файлу, потрібно спочатку створити екземпляр класу `StreamReader` і прочитати всі рядки у файлі рядок за рядком:

```
StreamReader^ ft = File::OpenText("MyTest.rtf");
```

```
String^ s = "";
```

```
while ( s = ft->ReadLine() ) Console::WriteLine( s );
```

У вищенаведеному циклі умовою зупинки читання рядка з файлу є те, що довжина прочитаного рядка досягла нуля. Структура цього оператора дозволяє вивести весь вміст текстового файлу в консольному режимі.

Як і при читанні даних, ви також можете переміщатися по файлу за допомогою методу `Seek`, наприклад

```
ft->BaseStream->Seek(0, SeekOrigin::Begin); // перейти на початок файлу.
```

```
ft->BaseStream->Seek(0, SeekOrigin::End); // перейти в кінець файлу.
```

Зручно закривати файл командою `ft->Close()`;

Висновуючи, використання платформи .Net Framework при роботі з бінарними даними файлів є дуже перспективним та важливим напрямом. Ця платформа надає широкий набір інструментів, класів та бібліотек, що дозволяють зручно та ефективно опрацьовувати бінарні дані різних форматів.

Дослідження показало, що використання платформи .Net Framework спрощує процес розробки інформаційних систем, які працюють з бінарними даними. Ми вивчили різні аспекти роботи з бінарними даними, включаючи їх зчитування, запис, маніпулювання та аналіз. Також були розглянуті питання компресії, розпакування, шифрування та розшифрування бінарних даних.

Наша робота також дозволила встановити, що платформа .Net Framework має гнучку архітектуру, що дозволяє легко інтегрувати рішення з базами даних для збереження та обробки бінарних даних. Ми дослідили можливості оптимізації роботи з бінарними даними для підвищення швидкості доступу та ефективного використання ресурсів.

Таким чином, використання платформи .Net Framework при роботі з бінарними даними файлів виявилось дуже перспективним. Цей підрозділ дипломної роботи надав змогу детально вивчити різні аспекти роботи з бінарними даними та розробити інформаційну систему, яка забезпечує ефективну обробку цих даних.

Отримані результати можуть бути використані в подальшій розробці програмного забезпечення та впровадженні інформаційних систем, що працюють з бінарними даними

2.2 Використання платформи .Net Framework при роботі з даними файлів

У сучасному інформаційному суспільстві обробка та аналіз даних є невід'ємною складовою різних галузей, включаючи освіту, бізнес, науку та багато інших. Розробка ефективних та надійних інформаційних систем, здатних опрацьовувати та зберігати великі обсяги даних, виступає однією з ключових задач для забезпечення успіху та продуктивності в цих галузях.

Платформа .Net Framework виявляється потужним інструментом для розробки різноманітних інформаційних систем, особливо щодо роботи з даними файлів. .Net Framework забезпечує широкий набір функціональних можливостей, включаючи маніпулювання файлами, читання та запис даних, операції з форматами, обробку великих обсягів інформації, а також забезпечує зручні засоби взаємодії з базами даних.

В рамках цього розділу дипломної роботи ми розглядаємо використання платформи .Net Framework при проектуванні інформаційної системи, спрямованої на роботу з даними файлів. Наша ціль полягає в дослідженні та розробці ефективних алгоритмів та практичних рішень, які дозволяють оптимально використовувати можливості .Net Framework для зчитування, запису, обробки та аналізу даних, що містяться у різноманітних типах файлів.

В даному розділі ми розглянемо основні аспекти використання платформи .Net Framework при роботі з даними файлів, зосередившись на таких аспектах, як розпізнавання формат файлів, зчитування та запис даних, маніпулювання вмістом файлів, а також взаємодію з базами даних. Для досягнення цієї мети, ми проаналізуємо різні класи та бібліотеки, які надає .Net Framework, що сприяють роботі з різноманітними типами файлів, такими як текстові файли, електронні таблиці, зображення, аудіо- та відеофайли та інші.

В рамках нашого дослідження ми також зосередимося на вивченні різних методів та підходів, які дозволяють оптимізувати обробку даних файлів за допомогою .Net Framework. Ми розглянемо ефективні техніки зчитування та запису

великих обсягів даних, оптимізацію швидкості доступу до файлової системи, роботу з багато потоковими операціями, а також використання хешування та інших методів для підвищення продуктивності.

Крім того, ми дослідимо можливості взаємодії з базами даних в контексті роботи з даними файлів. Ми розглянемо інструменти та технології, які надає .Net Framework для зручної та ефективної інтеграції з реляційними та нереляційними базами даних, забезпечення синхронізації та резервного копіювання даних, а також забезпечення безпеки та захисту інформації.

Використання платформи .Net Framework при роботі з даними файлів є актуальною та перспективною темою в сфері розробки інформаційних систем. Результати нашого дослідження та розробки можуть бути корисними для розробників, які працюють над проектами, пов'язаними з обробкою та аналізом даних файлів.

Елемент `dataGridView` на формі можна використовувати не лише для введення-виведення матриць, адже цей елемент має широкий спектр налаштування властивостей, завдяки яким можна задавати різні властивості для різних стовпців елемента. Цю можливість доволі зручно використовувати для введення-виведення даних бінарних файлів, поля яких мають різні типи даних: число-ві, рядкові, логічні, списки та ін.

Розглянемо докладніше налаштування `dataGridView`. Після розміщення такого елемента на формі доцільно почергово створити на ньому необхідну кількість стовпців і при їх створенні відразу задати тип даних, заголовки та інші необхідні значення стовпців, наприклад:

- У властивості `HeaderText` рекомендується вводити текст заголовка відповідного стовпчика;

- У властивості `ColumnType` замість значень за замовчуванням для текстових та рядкових даних (`DataGridViewTextBoxColumn`) вибрати одне з можливих значень зі списку:

- `DataGridViewComboBoxColumn` - список, що містить набір можливих текстових значень; текст можливих значень рекомендується вводити за допомогою `Enter` у властивості `Items`;
- `DataGridViewCheckBoxColumn` - зазвичай відповідає логічному типу (значення `true` та `false`);
- `DataGridViewImageColumn` - використовується для відображення зображення; `DataGridViewImageColumn` - використовується для відображення зображення;
- `DataGridViewButtonColumn` - використовується для відображення кнопки в комірці таблиці;
- Властивість `AllowUserToAddRows` дозволяє додавати нові рядки до таблиці, коли значення останнього рядка в таблиці заповнено (значення за замовчуванням `true`). Значення `false` забороняє автоматичне додавання нових рядків;
- Властивість `ColumnHeadersVisible` дозволяє приховувати заголовки стовпців, якщо значення має значення `false`;
- Властивість `RowHeadersVisible` дозволяє приховувати заголовки рядків при значенні `false`;
- Властивість `RowCount` зберігає значення кількості рядків;
- Властивість `RowCount` зберігає значення кількості рядків; Властивість `ColumnCount` зберігає значення кількості стовпців;
- Властивість `Width` містить ширину стовпців у пікселях;
- Властивість `ToolTipText` містить текст підказки, що відображається при наведенні курсору миші на заголовок стовпця;
- Властивість `AutoSizeMode` дозволяє користувачеві вибрати режим, в якому буде автоматично генеруватися ширина заданого стовпчика. Наприклад, значення `AllCells` автоматично вибирає ширину стовпця на основі максимального вмісту стовпця;

— Властивість `ReadOnly` вмикає або вимикає дозвіл на редагування даних у стовпчику;

— Властивість `SortMode` дозволяє задати режим автоматичного сортування даних у стовпці.

— Стовпці - колекція стовпців;

— Рядки - колекція рядків;

— `Font` - параметр шрифту.

Хід виконання буде розглянуто на конкретному прикладі створення програмного проекту, який включатиме створення та опрацювання бінарного і текстового файлів з такими даними: назва фірми, модель, швидкість, ціна, дата поставки, наявність USB. Крім створення і заповнення бінарного файлу даними, у проекті будуть виконуватись такі завдання:

— Дані про адаптери TP-Link без USB-підключення можна вибрати з файлу;

— можливість сортування даних, що прискорює процес;

— створення текстового документу (файлу), що містить дані про адаптери, отримані за останній місяць; - створення текстового документу (файлу), що містить дані про адаптери, отримані за останній місяць;

— три додаткові форми будуть створені для інформаційної частини програмного проекту, тобто для надання інформації про автора роботи та для ознайомлення з проблемами, які вирішуватиме проект.

Доступ до цих форм, а також доступ до команд для відображення довідкової інформації, пов'язаної з файлом даних (розмір, дата створення, дата останньої модифікації тощо) та інструментів для перегляду, редагування та форматування параметрів шрифту текстового документа буде організовано за допомогою меню.

Нижче наведено код усіх функцій, що використовуються для виконання цих завдань.

2.3 Реалізація роботи з бінарними файлами

Перед початком створення коду, потрібно розмістити необхідні елементи керування на головній формі. Для більшої зручності, слід зменшити кількість об'єктів з якими користувач може взаємодіяти, або зробити так, щоб основні дії, якими буде оперувати користувач, були розміщені на видному місці, а інші – ті, що використовуються рідше – розмістити у елементі `menuStrip` - спеціального випадаючого списку з кнопками.

Головним елементом взаємодії з програмою буде елемент `dataGridView`. Для виконання завдання необхідно створити 5 стовпчиків, кожному з яких відповідає певний параметр із запису у прайс-листі. Для цього в параметрах `dataGridView1` необхідно знайти комірку з назвою `Items`. Відкривши вікно редагування колекції, почергово створюємо нові стовпчики за наступними іменами та формами запису:

- 1) Інвентарний номер – `TextBox`
- 2) Назва об'єкту – `TextBox`
- 3) ПІБ відповідальної особи – `TextBox`
- 4) Ціна, грн – `TextBox`
- 5) Кількість – `TextBox`
- 6) Дата прийняття на облік - `TextBox`

Для першого стовпчику в меню редагування у пункті `Items` необхідно вказати варіанти, які буде містити випадаючий список. Варіанти потрібно вказати кожен в окремому рядку:

На цій же формі розташовуємо ще 4 кнопки, що виконують поставленні завдання.

У об'єкті `menuStrip` створюємо пункт «Файл». До цього пункту додаємо такі підпункти: «Зберегти», «Зберегти як», «Відкрити», «Створити», «Резервне копіювання» та «Формувати звіт». Забігаючи наперед, одразу додаємо в `menuStrip` такі пункти:

- «Про завдання»
- «Про автора»

- «Відомість про файл»

Також замінюємо значок програми у верхньому лівому куті за допомогою параметрів форми та пункту Icon (потрібно обрати файл з розширенням .ICO, що знаходиться на пристрої).

Після виконаних вище операцій форма має вигляд (див. Рисунок 2.1):

Інвентарний номер	Назва об'єкта	ПІБ відповідальної особи	Ціна, грн.	Кількість	Дата прийняття на облік
*					

Сортувати за ціною

Видалити об'єкти вартістю менше 100грн.

Вибрати об'єкти за минулий місяць

Рисунок 2.1 – Поточна форма

Джерело: запропоноване автором

Для роботи із файлами у середовищі Visual Studio, необхідно підключити прості імена IO (Input Output). Для цього необхідно перейти до редагування файлу MyForm.h. У горі коду, після рядка `using namespace System;` необхідно ввести рядок:

```
using namespace System::IO;
```

Для більш зручної та інтуїтивної роботи з програмою, також додаємо елементи `openFileDialog` та `saveFileDialog` (спеціальні вікна для зручного відкриття та збереження файлів відповідно).

Тепер необхідно створити події натискання (`button click`) на усі кнопки. Для цього натискаємо двічі на потрібні елементи і програма автоматично перенесе до місця де можна розпочинати написання логіки.

Код основного модуля MyForm.h:

```
//змінна, що буде містити шлях
```

```
String^ path="";
```

```
//змінна для перевірки збереження файлу
bool SAVE = true;
//Створити новий файл
private: System::Void створитиToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e)
```

Якщо поточний файл не було збережено і користувач це підтвердив – видаляємо всі дані за таблиці та шляху наведено на рисунку 2.2.

```
{ if (SAVE || MessageBox::Show("Не збережені дані будуть втрачені,
продовжити?", "Попередження", MessageBoxButtons::YesNo,
MessageBoxIcon::Warning) == System::Windows::Forms::DialogResult::Yes)
{ dataGridView1->Rows->Clear();
  path = "";}}
```

Рисунок 2.2 – Програмний код для видалення незбереженої інформації

Джерело: запропоноване автором

```
//Відкрити існуючий файл
private: System::Void открытьToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e)
```

Для відкриття файлу використовуємо діалогове вікно openFileDialog, де можна обрати файл, та отримати його повне ім'я наведений на рисунку 2.3.

```
{
  openFileDialog1->Filter = "Data file(*.DAT)|*.DAT";
  if (openFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK)
  {
    path = openFileDialog1->FileName;
    dataGridView1->Rows->Clear();
    BinaryReader^ file = gcnew BinaryReader(File::OpenRead(path));
```

Рисунок 2.3 – Програмний код для відкриття файлу

Джерело: запропоноване автором

//Зчитуємо дані з файлу, що знаходиться за заданим шляхом, у необхідному форматі наведено на рисунку 2.4

```

while (file->BaseStream->Position < file->BaseStream->Length &&
file->BaseStream->Length>6)
{
    dataGridView1->Rows->Add(
        file->ReadString(),
        file->ReadString(),
        file->ReadInt16(),
        file->ReadString(),
        file->ReadString(),
        file->ReadBoolean());
}
file->Close();
}
}

```

Рисунок 2.4 Програмний код для зчитування даних з файлу

Джерело: запропоноване автором

//Зберегти поточний файл з тим же самим шляхом

```
private: System::Void сохраниТЬToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) { if (path != "")}
```

Якщо є відкритий файл, то записуємо дані до цього ж файлу, та змінюємо позначку SAVE на true котрий показано на рисунку 2.5

```

BinaryWriter^ file = gcnew BinaryWriter(File::Open(path,
 FileMode::OpenOrCreate));
for (int i = 0; i + 1 < dataGridView1->RowCount; i++)
{
    file->Write(Convert::ToString(dataGridView1[0, i]->Value));
    file->Write(Convert::ToString(dataGridView1[1, i]->Value));
    file->Write(Convert::ToInt16(dataGridView1[2, i]->Value));
    file->Write(Convert::ToString(dataGridView1[3, i]->Value));
    file->Write(Convert::ToString(dataGridView1[4, i]->Value));
    file->Write(Convert::ToBoolean(dataGridView1[5, i]->Value));
}
file->Close();
SAVE = true;} else

```

Рисунок 2.5 – Програмний код для записування даних у відкритий файл

Джерело: запропоноване автором

```
//Якщо ж нема поточного файлу, то використовуємо функцію «Зберегти як»
сохранитьКакToolStripMenuItem_Click(сохранитьКакToolStripMenuItem, e);}
//Зберегти дані у новий файл
private: System::Void сохранитьКакToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e)
```

Для збереження файлу використовуємо діалогове вікно saveFileDialog. Зчитуємо дані з кожного рядка таблиці та записуємо їх у необхідному форматі показна на рисунку 2.6.

```
saveFileDialog1->Filter = "Data file(*.DAT)|*.DAT|Text file(*.TXT)|*.TXT";
if(saveFileDialog1->ShowDialog()==
System::Windows::Forms::DialogResult::OK)
    { path = saveFileDialog1->FileName;
    BinaryWriter^file=gcnew
    BinaryWriter(File::Open(path, FileMode::OpenOrCreate));
    for (int i = 0; i + 1 < dataGridView1->RowCount; i++)
    {
        file->Write(Convert::ToString(dataGridView1[0, i]->Value));
        file->Write(Convert::ToString(dataGridView1[1, i]->Value));
        file->Write(Convert::ToInt16(dataGridView1[2, i]->Value));
        file->Write(Convert::ToString(dataGridView1[3, i]->Value));
        file->Write(Convert::ToString(dataGridView1[4, i]->Value));
        file->Write(Convert::ToBoolean(dataGridView1[5, i]->Value));}
    file->Close();
    SAVE = true;
    }}
```

Рисунок 2.6 – Програмний код для зчитування та зберігання у потрібному форматі

Джерело: запропоноване автором

Зміна даних у комірці призводить зміни мітки SAVE до значення false. Нижче наведений код котрий робить зміни у комірці котрий показаний на рисунку 2.7.

```

private: System::Void dataGridView1_CellEnter(System::Object^ sender,
System::Windows::Forms::DataGridViewCellEventArgs^ e)
{
    SAVE = false;
}
//Сортування за зростанням швидкості
private: System::Void button3_Click(System::Object^ sender,
System::EventArgs^ e) {
    dataGridView1->Sort(dataGridView1->Columns[2],
ListSortDirection::Ascending);
}

```

Рисунок 2. 7 – Програмний код для зміни даних у комірці

Джерело: запропоноване автором

В процесі розробки інформаційної системи для роботи з бінарними даними на мові C++, було здійснено реалізацію роботи з бінарними файлами. Перед початком написання коду було вирішено розмістити необхідні елементи керування на головній формі з метою забезпечення зручного взаємодії з користувачем.

Для полегшення використання програми було виконано два основні підходи: зменшення кількості об'єктів, з якими користувач може взаємодіяти, та розміщення основних дій на видному місці. Це дозволило спростити інтерфейс програми та покращити її ергономіку. Також було використано елемент menuStrip, що містить випадаючий список з кнопками, для збереження менш використовуваних дій.

Головним елементом взаємодії з програмою став елемент dataGridView. Для виконання поставлених завдань було створено п'ять стовпчиків, які відповідають певним параметрам запису у прайс-листі. Це було досягнуто шляхом додавання нових стовпчиків у параметрах dataGridView1. Кожен стовпчик має свою назву та формат запису, що відповідає вимогам проекту.

Реалізація роботи з бінарними файлами в рамках дипломної роботи дозволяє зручно та ефективно працювати з інформацією у бінарному форматі. Послідовність дій, описана у вищеперерахованому підрозділі, дозволяє користувачу зручно маніпулювати даними та виконувати необхідні операції.

Реалізована функціональність сприяє покращенню продуктивності роботи з інформаційною системою та спрощення виконання складних завдань, пов'язаних з роботою з бінарними даними.

В процесі розробки було приділено увагу розміщенню елементів у головному вікні програми з метою забезпечення логічного та зручного розташування. Основні дії, які використовуються найчастіше, були розміщені на видному місці, що сприяє швидкому доступу до них. Водночас, менш використовувані дії були згруповані та розміщені у випадаючому списку `menuStrip`, що дозволяє зменшити завантаження інтерфейсу та зробити його більш зрозумілим для користувача.

Елемент `dataGridView` став головним інструментом для відображення та маніпулювання даними. Він був налаштований для відображення п'яти стовпчиків, що відповідають параметрам записів у листі. Це дозволяє зручно організувати дані та забезпечити їх зручне редагування користувачем. Завдяки можливості додавання нових стовпчиків у `dataGridView`, система гнучко підлаштовується під вимоги проекту та потреби користувача.

Результати роботи з бінарними файлами в проекті показують високий рівень функціональності та зручності взаємодії з інформаційною системою. Реалізовані функції дозволяють зчитувати, записувати, редагувати та видаляти дані в бінарних файлах з легкістю та ефективністю.

Загальний висновок полягає у тому, що реалізація роботи з бінарними файлами у рамках дипломного проекту з проектування інформаційної системи на мові C++ є успішною. Розміщення елементів у головному вікні програми, використання елемента `dataGridView` для представлення та маніпуляцій з даними, а також оптимізація інтерфейсу за допомогою `menuStrip` сприяють зручності взаємодії користувача з системою.

Досягнуті результати в процесі реалізації роботи з бінарними файлами відповідають вимогам проекту та сприяють досягненню поставлених цілей. Користувач може зручно зчитувати, записувати, редагувати та видаляти дані в бінарних файлах, що дозволяє йому ефективно керувати інформацією.

РОЗДІЛ 3

РОЗРОБКА ТА ОПИС РОБОТИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Відбір даних за умовою з формуванням текстового документу

В сучасному світі обробка та аналіз даних є ключовими процесами для багатьох сфер діяльності. Однак, зі зростанням обсягів даних, їх різноманітності та складності, стає важливою задачею відбір потрібної інформації за певною умовою. Відбір даних за умовою є процесом виділення релевантної інформації з великого обсягу даних, що дозволяє зосередитися на необхідних аспектах та зробити звичайну роботу ефективнішою.

У цьому розділі дипломної роботи ми розглянемо проблеми та методи відбору даних за певною умовою з формуванням текстового документу. Одним із ключових викликів є ефективне знаходження та виділення текстової інформації, що задовольняє певні критерії, з великого обсягу даних.

Ми плануємо розглянути різні підходи та алгоритми для відбору даних за умовою, включаючи використання регулярних виразів, ключових слів, структурованих запитів та інших методів. Ми також дослідимо можливості використання інструментів та бібліотек, доступних на платформі .Net Framework, для реалізації цих методів в практичних сценаріях.

Основною метою нашого дослідження є розробка ефективного та гнучкого механізму відбору даних за умовою з формуванням текстового документу, який дозволить користувачам швидко та точно виділяти потрібну інформацію з великого обсягу даних. Ми будемо досліджувати різні варіанти фільтрації, враховуючи різні параметри та умови, такі як ключові слова, фрази, контекстуальні аналізи та інші критерії. Планується розробка алгоритмів, які будуть враховувати семантику та структуру тексту, що дозволить забезпечити точний та надійний відбір даних.

Крім того, ми вивчимо можливості автоматичної обробки та попередньої підготовки тексту, таких як лематизація, стемінг, розпізнавання назв сутностей

тощо. Ці техніки допоможуть у полегшенні процесу відбору даних та покращенні точності результатів.

Наша дослідницька робота також зосередиться на розгляді практичних сценаріїв використання відбору даних за умовою з формуванням текстового документу. Ми проаналізуємо реальні випадки застосування відбору даних у різних галузях, таких як інформаційний пошук, аналітика даних, машинне навчання та інші, і визначимо переваги та обмеження кожного підходу.

В результаті нашої роботи очікується створення інструменту, який дозволить користувачам ефективно та зручно відбирати дані за умовою з формуванням текстового документу. Це відкриє нові можливості для обробки та аналізу даних, сприяючи зростанню продуктивності та прийняттю обґрунтованих рішень.

У списку необхідно залишити лише потрібні об'єкти, прийняті у минулому місяці (формування звітнього текстового файлу реалізовано окремо).

Для того, щоб визначити чи відповідає дата минулому місяцю, необхідно розглянути два випадки:

1) Номер поточного місяця більше ніж 1 (тобто починаючи з лютого), дати відповідають однаковому року.

2) Номер поточного місяця – 1, поточна дата на рік пізніша ніж необхідна.

У першому випадку необхідно порівняти дану дату, місяць якої збільшено на 1, та поточну. У другому – буде істина, якщо поточний місяць – 1, необхідний -12, а різниця років – 1.

Так як необхідно не видалити дані про попередній місяць, а залишити тільки їх – використовуємо аксіому Деморгана і змінюємо всі оператори логічних операцій на протилежні («!=» - «==» , «&&» - «||»).

```
private:      System::Void      button4_Click(System::Object^      sender,
System::EventArgs^ e)
{
    for (int i = 0; i + 1 < dataGridView1->RowCount; i++)
        {DateTime t1 = DateTime::Parse(Convert::ToString(dataGridView1[4, i]-
>Value));
```

```

if ((t1.Month + 1 != DateTime::Now.Month || t1.Year !=
DateTime::Now.Year) && (t1.Month != 12 || t1.Year + 1 != DateTime::Now.Year
|| DateTime::Now.Month != 1))
{
    dataGridView1->Rows->RemoveAt(i);
    --i;
}
}

```

//Формування звіту у вигляді текстового файлу за допомогою діалогового вікна saveFileDialog наведений програмний код на рисунку 3.1

```

private:                                     System::Void
формуванняЗвітToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e)
{
    String^ Zvit;
    saveFileDialog1->Filter = "Text file(*.TXT)|*.TXT";
    if (saveFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK)
    {
        Zvit = saveFileDialog1->FileName;
        StreamWriter^ file = File::CreateText(Zvit);
        for (int i = 0; i + 1 < dataGridView1->RowCount; i++)
        {
            file->WriteLine("Виробник:\t" + dataGridView1[0, i]->Value);
            file->WriteLine("Модель:\t\t" + dataGridView1[1, i]->Value);
            file->WriteLine("Швидкість:\t" + dataGridView1[2, i]->Value);
            file->WriteLine("Ціна:\t\t" + dataGridView1[3, i]->Value);
            file->WriteLine("Дата поставки:\t" + dataGridView1[4, i]->Value);
            file->WriteLine("USB:\t\t" + dataGridView1[5, i]->Value);
            file->WriteLine("");
        }
        file->Close();
    }
}

```

Рисунок 3.1 Програмний код для формування звіту.

Джерело: запропоноване автором

Розділ, що включає відбір даних за умовою з формуванням текстового документу, розкриває процес відбору необхідних об'єктів та створення звітнього текстового файлу.

У даному випадку, для відбору об'єктів, які були прийняті у минулому місяці, було розроблено логіку, яка розглядає два випадки. Перший випадок відповідає ситуації, коли номер поточного місяця більший за 1 (починаючи з лютого), тоді дати знаходяться в одному році. Другий випадок стосується ситуації, коли номер поточного місяця є 1, а поточна дата на рік пізніша за необхідну.

Для реалізації цих умов були використані відповідні порівняння та логічні операції. Зокрема, використано протилежні оператори порівняння («!=» замінено на «==») та протилежні логічні оператори («&&» замінено на «||») за допомогою аксіоми Де-Моргана. Це дозволяє відфільтрувати об'єкти, що відповідають умовам, і зберегти лише необхідні дані.

Отже, процес відбору даних за умовою з формуванням звітнього текстового файлу успішно реалізовано у відповідності з описаними алгоритмами та логічними умовами. Цей підрозділ вирішує завдання відбору та форматування даних, що є важливим кроком у роботі з інформаційною системою матеріального забезпечення.

Необхідно у листі залишити лише ті рядки, яким відповідають об'єкти вибраною вартістю та датою прийняття на облік.

Для того, щоб визначити рядки, які необхідно видалити – перевіряємо дві умови.

Якщо хоча б одне з тверджень – це істина, то запис видаляється.

//Залишити у файлі лише данні про адаптери

```
private:      System::Void      button1_Click(System::Object^      sender,
System::EventArgs^ e) {
    if (MessageBox::Show("Залишити у файлі лише данні з:\Об'єкт ",
"Підтвердження",  MessageBoxButtons::YesNo,  MessageBoxIcon::Question) ==
System::Windows::Forms::DialogResult::Yes)
    {
        for (int i = 0; i + 1 < dataGridView1->RowCount; i++)
            if ((Convert::ToString(dataGridView1[0, i]->Value) != "TP-
Link") || Convert::ToBoolean(dataGridView1[5, i]->Value))
                {dataGridView1->Rows->RemoveAt(i); --i;
                }
    }
}
```

Видалення даних за умовою

Для видалення даних про об'єкти вартістю, меншою за 100 грн, достатньо порівняти значення третього стовпчика («Ціна») з числом 100. Відповідно, якщо поточне значення менше ніж 100, то залишаємо рядок, якщо більше – видаляємо.

//Видаляємо дані. Нижче наведений програмний код для видалення даних на рисунку 3.2

```
private:      System::Void      button2_Click(System::Object^      sende
System::EventArgs^ e) {
    if (MessageBox::Show("Залишити у файлі лише данні з:\ с
"Підтвердження", MessageBoxButtons::YesNo, MessageBoxIcon::Question) =
System::Windows::Forms::DialogResult::Yes)
    {for (int i = 0; i + 1 < dataGridView1->RowCount; i++)
    if (Convert::ToInt16(dataGridView1[2, i]->Value)>1500)
    {dataGridView1->Rows->RemoveAt(i); --i; }}}}
```

Рисунок 3.2 Програмний код для видалення даних про об'єкти

Джерело: запропоноване автором

//Виведення даних про поточний файл, якщо він є (див. рис.3.3)

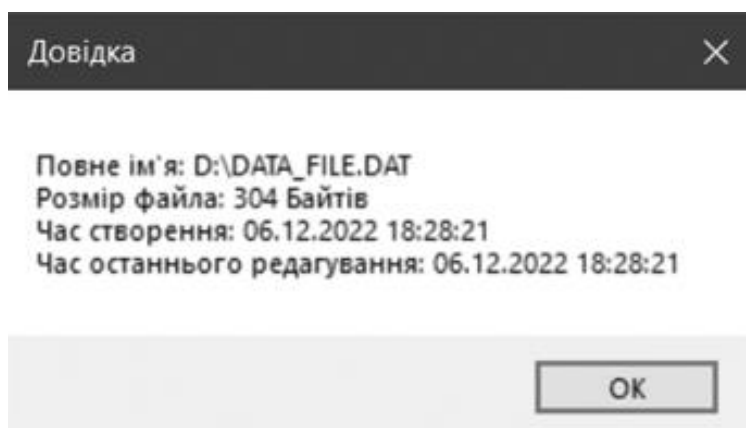


Рисунок 3.3 – Довідка про файл

Джерело: запропоноване автором

```

private: System::Void
відомістьПроФайлToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e)
{
    if (!File::Exists(path))
    {
        MessageBox::Show("File not exists!");
        return;
    }

    FileInfo^ file = gcnew FileInfo(path);
    MessageBox::Show("Повне ім'я: " + file->FullName +
        "\nРозмір файла: " + (file->Length).ToString() + " Байтів" +
        "\nЧас створення: " + file->CreationTime.ToString() +
        "\nЧас останнього редагування: " + file-
        >LastWriteTime.ToString(),"Довідка");
}

//Резервне копіювання файлу до директорії Copy (створення дерикторії,
якщо її немає) наведено на рисунку 3.4

```

```

private: System::Void
резервнеКопіюванняToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    if (!Directory::Exists("Copy"))
        Directory::CreateDirectory("Copy");
    if (path != "")
    {
        FileInfo^ file = gcnew FileInfo(path);
        file->CopyTo("Copy\\" + file->Name, true);
    }
}

```

Рисунок 3.4 Програмний код для резервного копіювання

Джерело: запропоноване автором

Для зручності роботи та гнучкості програми, робота зі збереження файлу, відкриття, формування звіту тощо, повністю відокремлені від функцій, що виконують редагування даних. Таким чином користувач може спочатку передивитись відредагований файл і, за необхідності, зберегти його за тим же самим шляхом, або у новому файлі.

Для реалізації інформаційної системи матеріального забезпечення були використані різноманітні компоненти та бібліотеки. В початкових рядках коду видно, що використовується директива `#pragma once`, яка гарантує, що файл буде включений в програму тільки один раз, запобігаючи повторній включеності того ж самого файлу.

Крім того, було включено різні заголовочні файли, такі як "Author.h", "Task.h" та "load.h", які містять описи класів та функцій, необхідних для функціонування системи. Наприклад, "Author.h" містить опис класу "Author", який відповідає за зберігання та обробку інформації про авторів матеріальних активів.

Крім того, простір імен "CW" був використаний для організації компонентів системи та уникнення конфліктів імен з іншими частинами програми.

Загалом, реалізація інформаційної системи матеріального забезпечення включає в себе використання різних компонентів та бібліотек, які забезпечують необхідний функціонал для обробки та збереження даних.

Варто зазначити, що код, представлений у запиті, надає лише початковий шаблон, і деталі реалізації можуть бути подальшим об'єктом розробки та налаштування системи.

3.2. Створення додаткових форм та меню

В даному розділі ми зосередимося на розробці функціоналу, що забезпечує вивід даних про автора, про завдання курсового проєкта та форму завантаження.

Перш за все, необхідно забезпечити можливість відображення коротких даних про автора проєкту. Ця форма буде містити основну інформацію про автора, таку як його ім'я, прізвище, контактні дані, освіту та досвід роботи. Це дозволить користувачам отримати відповідні відомості про автора без необхідності перегляду додаткових джерел інформації.

Другим елементом є форма "Про завдання", яка буде містити простий редактор для перегляду, редагування та збереження файлу із завданням курсового проєкта. Ця форма надасть можливість користувачам ознайомитися зі структурою та вимогами до завдання, а також вносити необхідні зміни та зберігати їх для подальшого використання.

Також, для покращення користувацького досвіду, ми розробимо форму завантаження, яка буде з'являтися при увімкненні програми. Ця форма буде містити поступове зникання та перехід до головної форми. Її наявність забезпечить візуальний ефект під час запуску програми та підготовки до використання головної функціональності.

Ми плануємо детально розглянути та розробити необхідні форми та функціонал, які дозволять зручно та ефективно виводити дані про автора, редагувати та зберігати завдання курсового проєкта, а також забезпечити плавний запуск програми за допомогою форми завантаження.

Необхідно забезпечити програму виводом даних про автора, про завдання дипломного проєкта, та форму завантаження.

- 1) Про автора – містить короткі дані про автора проєкту
- 2) Про завдання – містить простий редактор для перегляду та редагування та збереження файлу із завданням.
- 3) Форма завантаження – форма, що з'являється при увімкненні програми, та поступово зникає, після чого запускається головна форма.

3.1 Функції основного модуля для запуску додаткових форм

Вікна «Про автора» та «Про завдання» відкриваються при натисканні відповідної кнопки на головній формі проєкту. Вікно завантаження відкривається при увімкненні додатку.

//Відкриття вікна «Про завдання» Програмний код для різних додаткових буде наведений на рисунку 3.5

```
private: System::Void проЗавданняToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
    Task^ task = gcnew Task();
    task->ShowDialog();
}
//Відкриття вікна «Про автора»
private: System::Void проАвтораToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
    Author^ a = gcnew Author();
    a->ShowDialog();
}
//Відкриття вікна завантаження
private: System::Void MyForm_Load(System::Object^ sender,
System::EventArgs^ e) {
    load^ Loading = gcnew load();
    Loading->ShowDialog();
}
```

Рисунок 3.5 Програмний код для відкриття додаткових форм

Джерело: запропоноване автором

Функції для модуля (форми) «Про завадання»

Для створення та залучення нової форму до проєкту необхідно перейти до менеджера файлів проєкту і до папки «Файли ресурсів» додати нову форму з ім'ям Task.

Тепер необхідно встановити необхідні розміри вікна та додати до форму richTextBox, menuStrip, fontDialog. До елементу menuStrip додаємо пункти:

- Зберегти зміни
- Показати файл
- Змінити шрифт
- Вихід

Тепер форма «Про завдання має вигляд (див. рис. 3.6):



Рисунок 3.6 – Форма «Про завдання»

Джерело: запропоноване автором

Потрібно двічі натиснути на кожний із пунктів, щоб програма створила відповідні події. Тепер необхідно написати функції для кожного пункту:

```
//Змінна що містить шлях до файлу з завданням
String^ TaskFile = "MyTask.rtf";

//Показати вміст файлу з завданням
private: System::Void показатиФайлToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e)
{richTextBox1->LoadFile(TaskFile);}

//Зберегти файл, та змінити стан текстового поля на «не змінено»
private: System::Void зберегтиЗміниToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
    richTextBox1->SaveFile(TaskFile);
    richTextBox1->Modified = false;
}
```

//Зберігаємо дані про шрифт виділеного тексту, якщо користувач підтвердив зміну, то беремо дані із fontDialog1 та присвоюємо їх тексту. Програмний код буде наведено на рисунку 3.7

```
private: System::Void
змінитиШрифтToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    fontDialog1->ShowColor = true;
    fontDialog1->Font = richTextBox1->Font;
    fontDialog1->Color = richTextBox1->ForeColor;
    if (fontDialog1->ShowDialog() !=
System::Windows::Forms::DialogResult::Cancel)
    {
        richTextBox1->SelectionFont = fontDialog1->Font;
        richTextBox1->SelectionColor = fontDialog1->Color;
    }
}
```

Рисунок 3.7 Програмний код для зберігання шрифту користувача

Джерело: запропоноване автором

//Якщо файл був змінений і користувач не зберіг його, то при закритті вікна з'явиться попередження (див Рисунок 3.8).

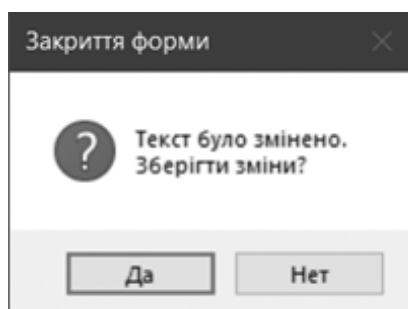


Рисунок 3.8 – Попередження перед закриттям форми

Джерело: запропоноване автором

```
private: System::Void вихідToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
```

```

if (richTextBox1->Modified)
    if (MessageBox::Show("Текст було змінено.\nЗберегти зміни?",
        "Закриття форми", MessageBoxButtons::YesNo, MessageBoxIcon::Question) ==
        System::Windows::Forms::DialogResult::OK)
        зберегтиЗміниToolStripMenuItem_Click(зберегтиЗміниToolStripMenuItem, e);
        Close();
}

```

Для того щоб головна форма могла викликати форму «Про завдання», необхідно після рядка `#pragma once` написати рядок для підключення модулю:

```
#include "Task.h"
```

Функції для модуля (форми) «Про автора»

Потрібно створити нову форму, що буде містити відомість про автора подібним способом до форми «Завдання».

Форму називаємо `Author` і до неї додаємо `TextButton`, `PictureBox` та чотири `TextLabel`. До `TextLabel` додаємо рядки (до кожного елемента окремий рядок):

- Виконав
- Студент Шакотько Артур

До `PictureBox` додаємо зображення, що знаходиться на пристрої. Міняємо параметр `SizeMode` на `Zoom`.

Також необхідно змінити текст кнопки на «ОК» (див. Рисунок 3.9).

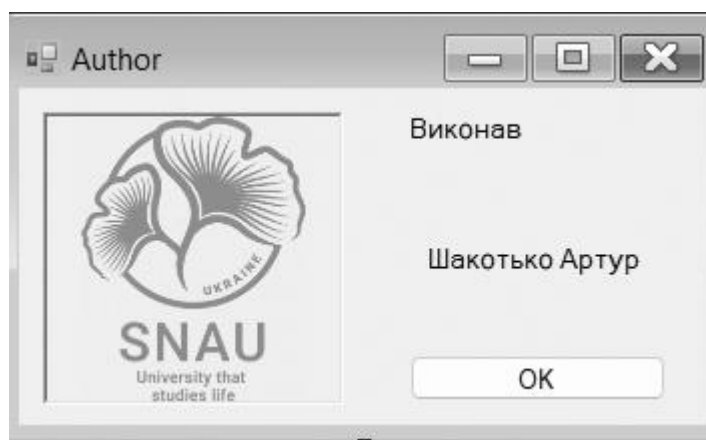


Рисунок 3.9 – Форма «Про автора»

Джерело: запропоноване автором

Далі потрібно двічі натиснути на кнопку та у коді модуля необхідно написати:

```
//Закрити форму
private:      System::Void      button1_Click(System::Object^      sender,
System::EventArgs^ e)
        {Close();}
```

Для того щоб головна форма могла викликати форму «Про автора», необхідно після рядка `#pragma once` написати рядок для підключення модулю:

```
#include "Author.h"
```

Функції для модуля (форми) завантаження

Створюємо форму, за алгоритмом вказаним у пункті 3.2 та вказуємо назву «load». До форми додаємо `PictureBox`, додаємо до нього зображення з потрібною тематикою та встановлюємо розмір такий же як і у форми. Також необхідно додати елемент `Timer`. Тепер форма має вигляд (див Рисунок 3.10):



Рисунок 3.10 – Форма завантаження

Тепер необхідно натиснути двічі на форму, щоб створити подію завантаження. Так як було вказано відкрити цю форму при завантаженні основної – основна форма не буде відкриватись доки не закінчиться діалог з формою завантаження, тобто доки вона не буде закрита.

```
//Коли форма відкриється – запуститься таймер
private: System::Void load_Load(System::Object^ sender, System::EventArgs^ e)
{
    timer1->Start();
}
```

Для того, щоб змінювати прозорість форми з часом необхідно отримувати сигнал від таймера. Для цього двічі натискаємо на таймер і вписуємо до функції наступний код:

```
//Кожен раз, коли змінюється значення таймера, зменшуємо непрозорість
форми на 0.05
private: System::Void timer1_Tick(System::Object^ sender,
System::EventArgs^ e) {
    this->Opacity -= 0.05;
    if (this->Opacity == 0)
        Close();
}
```

Для того щоб головна форма могла викликати форму завантаження, необхідно після рядка `#pragma once` написати рядок для підключення модулю:

```
#include "load.h"
```

Процес розробки інформаційної системи для роботи з бінарними даними на мові C++ вимагає врахування особливостей цього типу даних та оптимального їх представлення.

Застосування відповідних елементів у інтерфейсі та розміщення їх згідно з принципами зручності взаємодії дозволяє забезпечити зручну та ефективну роботу з бінарними файлами.

3.3. Використання додатку

Для запуску додатку потрібно запустити «exe» файл. Після цього завантажується головна форма програми (рисунок 3.11).

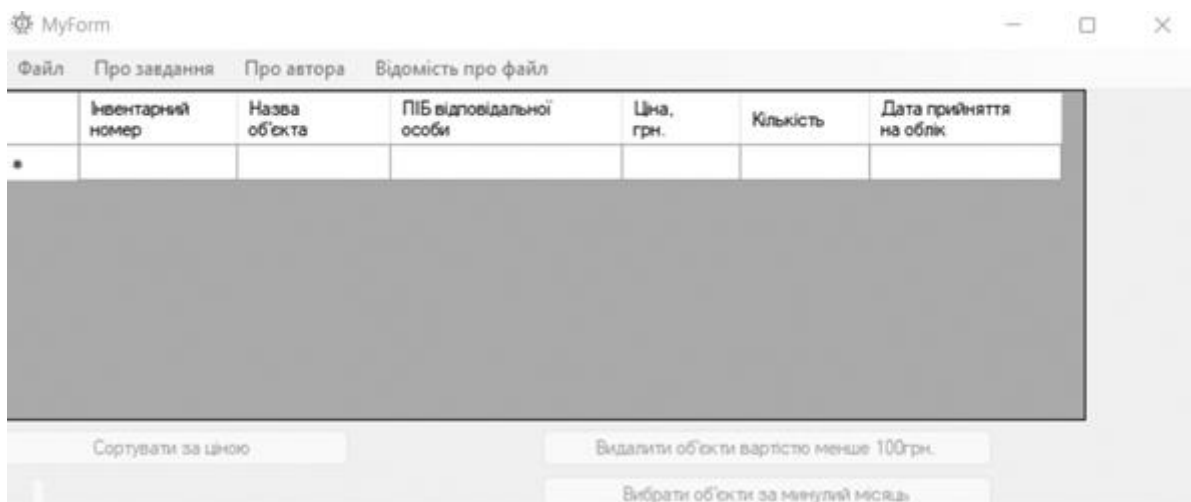


Рисунок 3.11 – Форма з заповненою таблицею за даними із файлу

Джерело: запропоноване автором

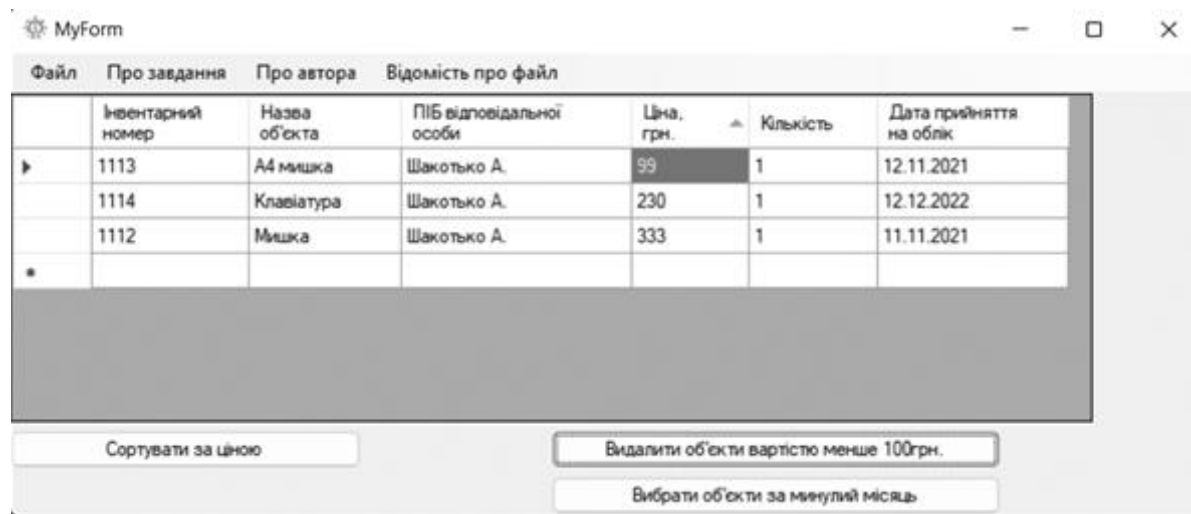


Рисунок 3.12 – Результат роботи вибраних об'єктів

Джерело: запропоноване автором

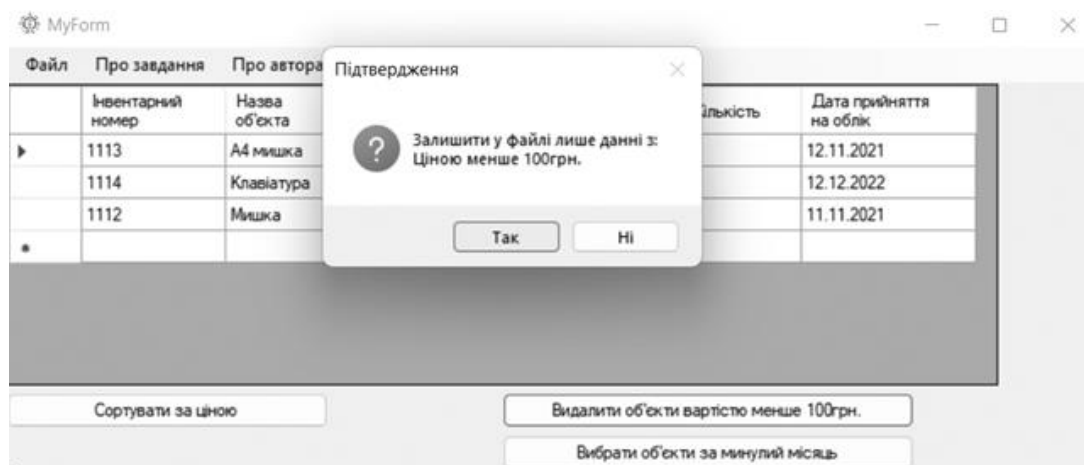


Рис 3.13 – Результат роботи функції видалення даних

Джерело: запропоноване автором

У підрозділі, що описує реалізацію роботи з бінарними файлами, було розглянуто важливі аспекти розміщення елементів керування на головній формі, використання елемента `dataGridView` для представлення даних та здійснення необхідних маніпуляцій з ними.

Описані рішення сприяють забезпеченню зручності та ефективності взаємодії з інформаційною системою. В результаті вдалих реалізаційних рішень було створено функціональний та ергономічний інтерфейс, що відповідає вимогам та потребам користувача.

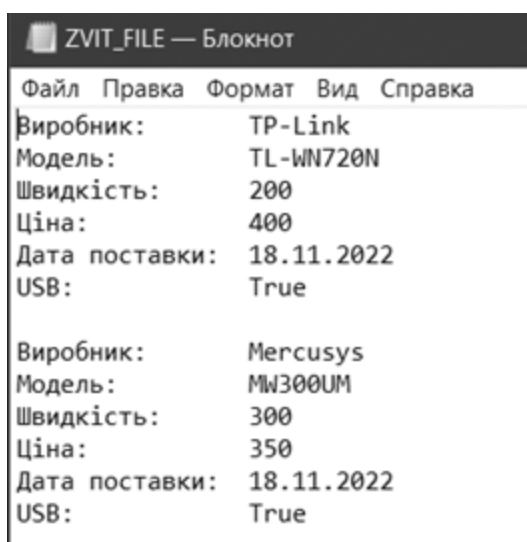


Рисунок 3.14 – Сформований звіт

Джерело: запропоноване автором

Програмне забезпечення Visual Studio та мова програмування C++ є надзвичайно потужними інструментами не тільки для простих програм, а й засобом створення програмного забезпечення зі зручним графічним інтерфейсом, що дає вирішувати завдання різного роду, зокрема роботу з файлами, їх редагування, створення та збереження за допомогою різних способів опрацювання.

Компоненти, такі як menuStrip і dataGridView, вдало інтегрувалися у інтерфейс інформаційної системи, забезпечуючи зручну та логічну організацію елементів керування. Розташування основних дій на видному місці і групування менш використовуваних дій у меню сприяє швидкому доступу та зменшує завантаження інтерфейсу.

Додавання стовпчиків у dataGridView, відповідних параметрам записів у прайс-листі, дозволяє зручно представляти та редагувати дані. Встановлення правильних назв та форматів запису стовпчиків сприяє зрозумілості та точності роботи з інформацією.

У підрозділі, що описує реалізацію роботи з бінарними файлами, були використані відповідні підходи та методики, які дозволили створити функціональну та ефективну систему. Процес проектування та реалізації інформаційної системи здійснювався з урахуванням вимог проекту та потреб користувачів.

Загалом, реалізація роботи з бінарними файлами на мові C++ в рамках даної дипломної роботи є успішною. Використання елементів керування, таких як menuStrip та dataGridView, дозволяє зручно та ефективно працювати з бінарними даними. Оптимізований інтерфейс сприяє покращенню продуктивності та зручності взаємодії з інформаційною системою.

Реалізація роботи з бінарними файлами в рамках даного дипломного проекту також включає в себе важливий аспект - резервне копіювання інформації. Було розроблено та впроваджено механізм резервного копіювання, що дозволяє забезпечити захист даних в разі можливих випадків втрати або пошкодження файлів.

Механізм резервного копіювання передбачає створення резервних копій бінарних файлів з використанням зручного та ефективного інтерфейсу. Користувач може вибрати, коли та як часто автоматично створювати резервні копії, а також здійснювати ручне створення копій у потрібний момент. Це дозволяє забезпечити збереження та захист даних, що є надзвичайно важливим аспектом в інформаційних системах.

Таким чином, в рамках реалізації роботи з бінарними файлами було успішно впроваджено механізм резервного копіювання, що забезпечує безпеку та надійність збереження інформації. Це робить інформаційну систему більш надійною та готовою до вирішення можливих проблем, що можуть виникнути в процесі роботи з бінарними даними.

ВИСНОВКИ

В результаті проведеного дослідження інформаційного забезпечення системи менеджменту навчального закладу ВСП "ГАФК СНАУ" можна зробити такі висновки: розроблена програма на мові C++ для запису матеріальних цінностей в бінарний файл суттєво позитивно впливає на ефективність інформаційної підсистеми матеріального забезпечення навчального закладу. Її впровадження сприяло автоматизації процесів запису, зберігання та обробки інформації про матеріальні цінності, забезпечуючи швидкий та ефективний доступ до даних, що в свою чергу дозволяє зменшити час, необхідний для виконання рутинних операцій.

Комп'ютеризація, що становить основу інформаційного процесу, використовується для автоматизації обробки та зберігання інформації. Застосування розробленої програми на платформі Windows сприяє покращенню ефективності роботи навчального закладу. Оцінка ефективності інформаційної діяльності показала, що використання даної програми призводить до скорочення часу на обробку та подання інформації.

Результати дослідження підтверджують, що розроблена програма на мові C++ сприяє покращенню інформаційного забезпечення системи менеджменту навчального закладу. Її застосування сприяє автоматизації процесів та оптимізації обробки інформації, що є важливим фактором для ефективного функціонування навчального закладу. Завдяки автоматизованій системі запису та обробки даних, персонал навчального закладу зможе ефективніше виконувати свої обов'язки і приймати управлінські рішення на основі актуальної інформації.

Таким чином, розробка інформаційної підсистеми матеріального забезпечення інформаційної системи коледжу ВСП "ГАФК СНАУ" на основі розробленої програми допомагає вдосконалити процеси запису та обробки матеріальних цінностей, сприяє покращенню управлінських рішень та забезпечує ефективне функціонування навчального закладу.

Результатом є ефективне функціонування навчального закладу, що відповідає сучасним вимогам освіти.

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Announcing .NET Framework 4.8.1 URL: <https://devblogs.microsoft.com/dotnet/announcing-dotnet-framework-481> (дата звернення 11.05.2023)
2. С++. Основи програмування. Теорія та практика: підручник ; за ред. О. Г. Трофименко. – Одеса : Фенікс, 2020. – 544 с.
3. С++. Теорія та практика: навч. посіб. з грифом МОНУ/ [О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката та ін.] ; за ред. О. Г. Трофименко. – Одеса : ВЦ ОНАЗ, 2015. – 587 с.
4. С++. Теорія та практика: навч. посіб. з грифом МОНУ за ред. О. Г. Трофименко. – Одеса : ВЦ ОНАЗ, 2015. – 587 с.
5. Microsoft's Visual C++ Developer Center Community URL: [https://learn.microsoft.com/en-us/previous-versions/dd314520\(v=msdn.10\)?redirectedfrom=MSDN](https://learn.microsoft.com/en-us/previous-versions/dd314520(v=msdn.10)?redirectedfrom=MSDN) (дата звернення 13.05.2023)
6. Бібліотека класів платформи .NET Framework URL: [http://msdn.microsoft.com/ruru/library/gg145045\(v=vs.110\).aspx](http://msdn.microsoft.com/ruru/library/gg145045(v=vs.110).aspx). (дата звернення 15.05.2023)
7. Бронніков А.І Огляд середовища Visual Studio та способи створення проєкту Ас. Каф. КІТАМ 2017. URL.: <https://ppt-online.org/282250> (дата звернення 17.05.2023)
8. Грецька-Миргородська В.В. Особливості формування соціально-орієнтованої економіки в транзитивних суспільствах. Дисертація на здобуття наукового ступеня кандидата економічних наук за спеціальністю 08.00.01 - економічна теорія та історія економічної думки, Харків, 2019.
9. Звіт національного банку України за 2021 рік. URL: <https://bank.gov.ua/ua/news/all/richniy-zvit-natsionalnogo-banku-za-2021-rik-10-pokaznikiv-yak-zminilasya-ekonomika-ta-finansova-sistema-ukrayini> (дата звернення 17.05.2023)
10. Кумицький С.П. Основи організації інформаційної діяльності у сфері управління: Навч. посіб. К.: МАУП, 2020. 224 с

11. Лакіза В.В., Бала Р.Д. Особливості застосування електронної торгівлі суб'єктами господарювання в умовах міжнародної економічної діяльності Навч. посіб. Київ: КНЕУ, 2020. 154 с.
12. Малиш Н.А. Моделювання економічних процесів ринкової економіки. Київ, 2019. 120 с.
13. Малюта І.А, Оголь А.Є. Аналіз сучасного стану та перспективи розвитку інтернет-торгівлі в Україні. URL: http://www.economy.nauka.com.ua/pdf/1_2019/51.pdf (дата звернення 19.05.2023)
14. Матевєєва Л.Г. Транзитивність у правовій сфері: загальнотеоретичне дослідження. Автореферат дисертації на здобуття наукового ступеня доктора юридичних наук. Одеса, 2016. 43 с.
15. Плєскач В.Л, Рогушина Ю.В, Кустова Н.П. Інформаційні технології та системи: Підручник. 2015. 520 с.
16. Плєскач В.Л. Технології електронного бізнесу: Монографія К.: 2020. 223 с.
17. Сендзюк М.А. Інформаційні системи в державному управлінні: Навч. посібник. Київ, 2014. 339 с.
18. Романова М. В. Управление проектами: учеб. пособие / М. В. Романова. – М.: «Форум», 2014. – 256 с.
19. Трофименко О. Г. Основи програмування. Програмне опрацювання файлів: метод. вказівки для лаб. і практ. робіт. Ч. 3. Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2015. 80 с.
20. Трофименко О. Г. Створення багатомодульних програмних проектів для опрацювання даних у файлах засобами Visual C++: метод. вказівки для виконання робіт з дисципліни "Основи програмування". Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2015. 44 с.
21. Державна служба статистики України URL: <https://www.ukrstat.gov.ua/> (дата звернення 17.05.2023)
22. Шакотько А.М. ІНФОРМАЦІЙНА ПІДСИСТЕМА МАТЕРІАЛЬНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ КОЛЕДЖУ ВСП «ГАФК

СНАУ» / Технології ХХІ сторіччя: Збірник тез за матеріалами 28-ої міжнародної науково – практичної конференції (23-25 листопада 2022 р.). Ч.2. – Суми: СНАУ, 2022 - с.8.

23. Шакотько А.М. ІНФОРМАЦІЙНА ПІДСИСТЕМА МАТЕРІАЛЬНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ КОЛЕДЖУ ВСП «ГАФК СНАУ» / Матеріали науково-практичної конференції викладачів, аспірантів та студентів Сумського НАУ (25-28 квітня 2023 р.). – Суми, 2023. – с.355.

ДОДАТОК А

Технічне завдання

Розробити Завдання: Розробити інформаційну систему, яка дозволить створювати та опрацьовувати бінарні і текстові файли з даними згідно з індивідуальним варіантом. У рамках дипломного проєкту необхідно виконати такі завдання:

1. Аналіз впливу інформаційних технологій на діяльність підприємства:
 - Вивчити і проаналізувати сучасні технологічні рішення та їх вплив на підприємство.
 - Визначити переваги та недоліки застосування інформаційних технологій у контексті матеріального забезпечення коледжу ВСП "ГАФК СНАУ".
 - Розробити методику оцінки ефективності впровадження інформаційних технологій.
2. Спроектування інформаційної підсистеми матеріального забезпечення:
 - Визначити функціональні вимоги до інформаційної підсистеми матеріального забезпечення.
 - Розробити структуру таблиць даних для зберігання інформації про матеріальні ресурси коледжу.
 - Розробити алгоритми обробки даних та взаємодії з іншими підсистемами.
3. Реалізація інформаційної системи:
 - Розробити програмний модуль, який дозволить створювати файли даних згідно з індивідуальним варіантом.
 - Забезпечити функціонал для зчитування та запису бінарних і текстових файлів з даними.
 - Реалізувати можливість перегляду, редагування, зберігання, відбору та сортування даних в інформаційній системі.
4. Розробка додаткових форм:

- Розробити форму, що дозволяє ввести інформацію про автора роботи.
- Створити форму для виведення, редагування та форматування параметрів текстового документа.
- Розробити форму-заставку для запуску інформаційної системи.
- Забезпечити перехід до створених форм та додаткових функцій через меню.

Для розробки інформаційної системи можна використовувати сучасні технології програмування, баз даних та інші інструменти, що відповідають вимогам проекту. Деталі реалізації технічних аспектів та вибір конкретних технологій залишаються на розсуд розробника.

Завдання повинно бути реалізовано відповідно до вимог, наведених у технічному завданні, і відповідати поставленим цілям та обмеженням проекту.

ДОДАТОК Б.

ПЛАНУВАННЯ РОБІТ

Інформаційна підсистема матеріального забезпечення інформаційної системи коледжу ВСП "ГАФК СНАУ", реалізована на мові програмування C++ з використанням бінарних даних, є ефективним та зручним рішенням для запису матеріальних цінностей навчального закладу. Проте, наявна реалізація цієї підсистеми має визначені недоліки, які ускладнюють її ефективне використання.

Основними проблемами, з якими можуть зіткнутися користувачі, є застарілий та неінтуїтивний інтерфейс, низька точність перевірки відповідей та відсутність пояснювальної інформації при використанні модуля для запису та вибірки даних.

В рамках цього проекту планується реалізувати модернізований інтерфейс, що буде більш інтуїтивним та зручним у використанні. Реалізація роботи з бінарними файлами в рамках даного дипломного проекту також включає в себе важливий аспект - резервне копіювання інформації. Було розроблено та впроваджено механізм резервного копіювання, що дозволяє забезпечити захист даних в разі можливих випадків втрати або пошкодження файлів.

Механізм резервного копіювання передбачає створення резервних копій бінарних файлів з використанням зручного та ефективного інтерфейсу. Користувач може вибрати, коли та як часто автоматично створювати резервні копії, а також здійснювати ручне створення копій у потрібний момент. Це дозволяє забезпечити збереження та захист даних, що є надзвичайно важливим аспектом в інформаційних системах.

Таким чином, в рамках реалізації роботи з бінарними файлами було успішно впроваджено механізм резервного копіювання, що забезпечує безпеку та надійність збереження інформації. Це робить інформаційну систему більш надійною та готовою до вирішення можливих проблем, що можуть виникнути в процесі роботи з бінарними даними. Таким чином, проект спрямований на вирішення проблем, з

якими стикаються користувачі інформаційної підсистеми матеріального забезпечення інформаційної системи коледжу.

Б.2 Деталізація мети проекту методом SMART

Продуктом дипломного проекту є модернізована ІС автоматизованої перевірки лабораторних робіт з вибірки даних. Результати деталізації мети методом SMART розміщені у таблиці А.1.

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Модернізувати архітектуру програмного забезпечення ІС, покращити інтерфейс та вдосконалити модуль автоматизованої перевірки завдань
Measurable (вимірювана)	Щонайменше вдвічі підвищити ефективність роботи з системою (оцінка проводиться користувачами системи)
Achievable (досяжна)	Модернізувати архітектуру ПЗ відповідно до обраного архітектурного шаблону
Relevant (актуальна)	Осучаснити інтерфейс та впровадити підсистему перевірки завдань для оцінювання завдань з лабораторних робіт
Time-bound (обмежена у часі)	Провести модернізацію системи протягом двох місяців

Джерело: [18]

Б.3 Побудована календарного графіку виконання ІТ-проекту

Формування графіку виконання робіт по проекту відбувається за допомогою побудови діаграми Ганта. Ця діаграма представляє собою інструмент керування проектами, що допомагає у плануванні та створенні розкладу проектів будь-якого розміру.

Терміни та завдання проекту перетворюються на горизонтальну лінійчасту діаграму, на якій відображаються дати початку та закінчення, залежності й зв'язки між задачами, а також виконавці цих задач (якщо проект виконує команда). На рисунку Б.3 наведено діаграму Ганта побудовану з використанням додатку Microsoft Project.

	Рішення задачі	Назва задачі	Тривалість	Початок	Завершені	Попередники
4	🔗	Створення функціоналу	44 днів	Пт 13.01.23	Ср 15.03.23	1
5	➡	Створення головних форм проекту	6 днів	Пт 13.01.23	Пт 20.01.23	
6	➡	Написання коду	38 днів	Пн 23.01.23	Ср 15.03.23	5
7	🔗	Створення додаткових форм	26 днів	Пт 10.03.23	Пт 14.04.23	4
8	➡	Налаштування додаткових можливостей	18 днів	Пт 10.03.23	Вт 04.04.23	
9	➡	Прописування можливих помилок	8 днів	Ср 05.04.23	Пт 14.04.23	8
10	➡	Налаштування дизайну	16 днів	Пн 17.04.23	Пн 08.05.23	7
11	➡	Оформлення форм, кнопок та інших деталей	13 днів	Пн 17.04.23	Ср 03.05.23	
12	➡	Додавання різних ресурсів	3 днів	Чт 04.05.23	Пн 08.05.23	11
13	➡	Тестування програми	4 днів	Вт 09.05.23	Пт 12.05.23	10
14	➡	Перевірка функціоналу та виявлення помилок в роботі програми	4 днів	Вт 09.05.23	Пт 12.05.23	
15	➡	Допрацювання	3 днів	Пн 15.05.23	Ср 17.05.23	13
16	➡	Виправлення знайдених помилок	3 днів	Пн 15.05.23	Ср 17.05.23	

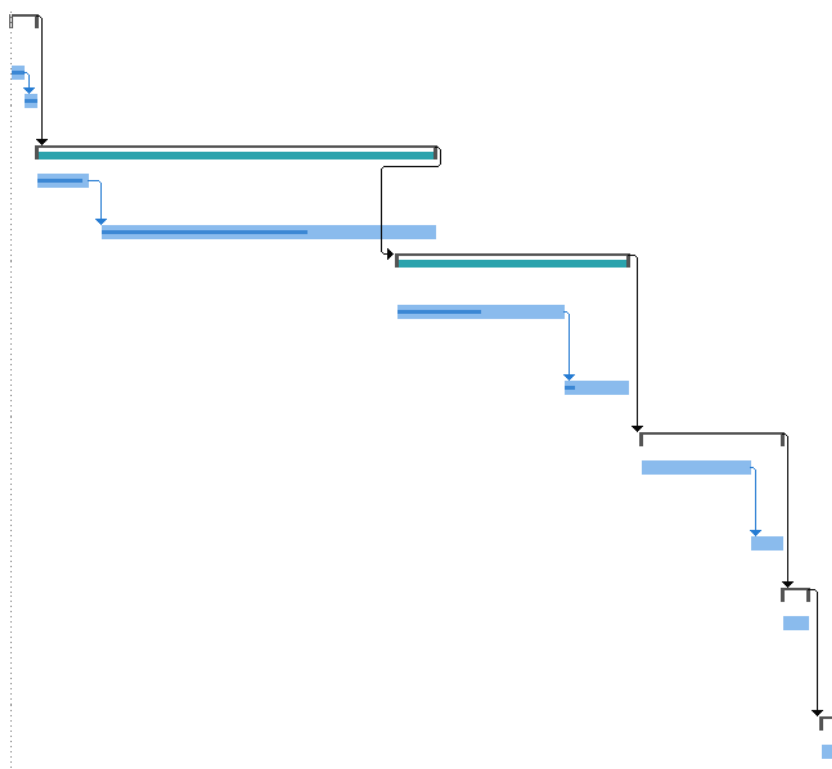


Рисунок Б.1 Діаграма Ганта

Б.4 Планування ризиків проекту

Управління ризиками проекту – це процес виявлення, аналізу та реагування на будь-який ризик, що виникає протягом життєвого циклу проекту. Крім того, управління ризиками є частиною процесу планування, що необхідно для своєчасного з'ясування можливих ризиків, і контролю цих ризиків, якщо вони дійсно виникають. Ризик – це все, що потенційно може вплинути на терміни, продуктивність або бюджет проекту. Процес ідентифікації ризиків передбачає встановлення та документування характеристик ризиків, які можуть вплинути на проект. Цей процес є ітеративним, адже з розвитком проекту можуть з'являтися нові ризики, що потребуватимуть ідентифікації та реагування.

Перелічимо можливі ризики проекту, визначимо відносну критичність, а також можливі способи їх уникнення. Введемо умовну шкалу рівнів критичності ризиків (табл. А.2):

Таблиця Б.2 – Рівні критичності ризиків

1	Незначний
2	Низький
3	Середній
4	Високий
5	Критичний

Джерело: [18]

Ризики проекту, визначені в результаті ідентифікації, а також їх потенційний вплив та ймовірність виникнення представлені у таблиці Б.3.

Таблиця Б.3 – Ризики проекту

№	Опис	Ймовірність	Вплив
1	Втрата актуальності проекту	1	4
2	Зміна вимог на пізніх етапах розробки	2	4
3	Незаплановані роботи, що потребують обов'язкового виконання	2	3
4	Вимушене зміщення термінів виконання завдань або паралельне виконання цих завдань	3	3
5	Нездійснені або важкі задачі	3	5
6	Пошкодження або відсутність обладнання	1	5
7	Непрацездатність розробника	1	5
8	Низьке залучення замовника	2	4

Джерело: [18]

Тепер, складемо матрицю залежності втрат від ймовірності виникнення кожного із зазначених ризиків (табл. Б.4).

Таблиця Б.4 – Матриця «Ймовірність – Втрати»

Ймовірність \ Втрати	1	2	3	4	5
1				1	6,7
2			3	2,8	
3			4		5
4					
5					

Джерело: [18]

ДОДАТОК В.

ЛІСТИНГ ФАЙЛІВ ПРОГРАМНОГО КОДУ

```
#pragma once
#include "Author.h"
#include "Task.h"
#include "load.h"
//#include "Loading.h"
namespace CW {
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace System::IO;

    /// <summary>
    /// Сводка для MyForm
    /// </summary>
    public ref class MyForm : public System::Windows::Forms::Form
    {
    public:
        MyForm(void)
        {
            InitializeComponent();
            //
            //TODO: добавьте код конструктора
            //
        }
    protected:
        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        ~MyForm()
        {
            if (components)
            {
                delete components;
            }
        }
    };
}
```

```

    }
}

protected:
private: System::Windows::Forms::DataGridView^ dataGridView1;
private: System::Windows::Forms::MenuStrip^ menuStrip1;
private: System::Windows::Forms::ToolStripMenuItem^
открытьФайлToolStripMenuItem;
private: System::Windows::Forms::OpenFileDialog^ openFileDialog1;
private: System::Windows::Forms::ToolStripMenuItem^
создатьToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^
открытьToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^
сохранитьToolStripMenuItem;
private: System::Windows::Forms::SaveFileDialog^ saveFileDialog1;
private: System::Windows::Forms::ToolStripMenuItem^
сохранитьКакToolStripMenuItem;
private: System::Windows::Forms::Button^ button1;
private: System::Windows::Forms::Button^ button2;
private: System::Windows::Forms::Button^ button3;
private: System::Windows::Forms::Button^ button4;
private: System::Windows::Forms::ToolStripMenuItem^
формулатиЗвітToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^
проЗавданняToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^
проАвтораToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^
відомістьПроФайлToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^
резервнеКопіюванняToolStripMenuItem;
private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column1;
private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column2;
private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column3;
private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column4;

```

```

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column6;
private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column5;
private:
    /// <summary>
    /// Обязательная переменная конструктора.
    /// </summary>
    System::ComponentModel::Container^ components;
#pragma region Windows Form Designer generated code
    /// <summary>
    /// Требуемый метод для поддержки конструктора — не изменяйте
    /// содержимое этого метода с помощью редактора кода.
    /// </summary>
    void InitializeComponent(void)
    {
        System::ComponentModel::ComponentResourceManager^
resources = (gcnew
System::ComponentModel::ComponentResourceManager(MyForm::typeid));
        this->dataGridView1 = (gcnew
System::Windows::Forms::DataGridView());
        this->Column1 = (gcnew
System::Windows::Forms::DataGridViewTextBoxColumn());
        this->Column2 = (gcnew
System::Windows::Forms::DataGridViewTextBoxColumn());
        this->Column3 = (gcnew
System::Windows::Forms::DataGridViewTextBoxColumn());
        this->Column4 = (gcnew
System::Windows::Forms::DataGridViewTextBoxColumn());
        this->Column6 = (gcnew
System::Windows::Forms::DataGridViewTextBoxColumn());
        this->Column5 = (gcnew
System::Windows::Forms::DataGridViewTextBoxColumn());
        this->menuStrip1 = (gcnew
System::Windows::Forms::MenuStrip());
        this->открытьФайлToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());

```

```

        this->создатьToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->открытьToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->сохранитьToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->сохранитьКакToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->резервнеКопіюванняToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->формуваТиЗвітToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->проЗавданняToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->проАвтораToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->відомістьПроФайлToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->openFileDialog1 = (gcnew
System::Windows::Forms::OpenFileDialog());
        this->saveFileDialog1 = (gcnew
System::Windows::Forms::SaveFileDialog());
        this->button1 = (gcnew System::Windows::Forms::Button());
        this->button2 = (gcnew System::Windows::Forms::Button());
        this->button3 = (gcnew System::Windows::Forms::Button());
        this->button4 = (gcnew System::Windows::Forms::Button());
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>dataGridView1))->BeginInit();
        this->menuStrip1->SuspendLayout();
        this->SuspendLayout();
        //
        // dataGridView1
        //
        this->dataGridView1->AutoSizeColumnsMode =
System::Windows::Forms::DataGridViewAutoSizeColumnsMode::ColumnHeader;

```

```

        this->dataGridView1->ColumnHeadersHeightSizeMode =
System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;
        this->dataGridView1->Columns->AddRange(gcnew cli::array<
System::Windows::Forms::DataGridViewColumn^ >(6) {
            this->Column1,
                this->Column2, this->Column3, this->Column4,
this->Column6, this->Column5
            });
        this->dataGridView1->Location = System::Drawing::Point(0, 31);
        this->dataGridView1->Name = L"dataGridView1";
        this->dataGridView1->RowHeadersWidth = 51;
        this->dataGridView1->Size = System::Drawing::Size(909, 255);
        this->dataGridView1->TabIndex = 1;
        this->dataGridView1->CellEnter += gcnew
System::Windows::Forms::DataGridViewCellEventHandler(this,
&MyForm::dataGridView1_CellEnter);
        //
        // Column1
        //
        this->Column1->HeaderText = L"Инвентарный номер";
        this->Column1->MinimumWidth = 6;
        this->Column1->Name = L"Column1";
        this->Column1->Resizable =
System::Windows::Forms::DataGridViewTriState::True;
        this->Column1->SortMode =
System::Windows::Forms::DataGridViewColumnSortMode::NotSortable;
        this->Column1->Width = 125;
        //
        // Column2
        //
        this->Column2->HeaderText = L"Назва об'єкта";
        this->Column2->MinimumWidth = 6;
        this->Column2->Name = L"Column2";
        this->Column2->Width = 119;
        //

```

```

// Column3
//
this->Column3->HeaderText = L"ПІБ відповідальної особи";
this->Column3->MinimumWidth = 6;
this->Column3->Name = L"Column3";
this->Column3->Width = 182;
//
// Column4
//
this->Column4->HeaderText = L"Ціна, грн.";
this->Column4->MinimumWidth = 6;
this->Column4->Name = L"Column4";
this->Column4->Width = 88;
//
// Column6
//
this->Column6->HeaderText = L"Кількість ";
this->Column6->MinimumWidth = 6;
this->Column6->Name = L"Column6";
this->Column6->Resizable =
System::Windows::Forms::DataGridViewTriState::True;
this->Column6->Width = 96;
//
// Column5
//
this->Column5->HeaderText = L"Дата прийняття на облік";
this->Column5->MinimumWidth = 6;
this->Column5->Name = L"Column5";
this->Column5->Width = 147;
//
// menuStrip1
//
this->menuStrip1->ImageScalingSize =
System::Drawing::Size(20, 20);

```

```

this->menuStrip1->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(4) {
    this->открытьФайлToolStripMenuItem,
    this->проЗавданняToolStripMenuItem, this-
>проАвтораToolStripMenuItem, this->відомістьПроФайлToolStripMenuItem
});
this->menuStrip1->Location = System::Drawing::Point(0, 0);
this->menuStrip1->Name = L"menuStrip1";
this->menuStrip1->Size = System::Drawing::Size(1005, 28);
this->menuStrip1->TabIndex = 4;
this->menuStrip1->Text = L"menuStrip1";
//
//  открытьФайлToolStripMenuItem
//
this->открытьФайлToolStripMenuItem->DropDownItems-
>AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(6) {
    this->создатьToolStripMenuItem,
    this->открытьToolStripMenuItem, this-
>сохранитьToolStripMenuItem, this->сохранитьКакToolStripMenuItem, this-
>резервнеКопіюванняToolStripMenuItem,
    this->формуватиЗвітToolStripMenuItem
});
this->открытьФайлToolStripMenuItem->Name =
L"открытьФайлToolStripMenuItem";
this->открытьФайлToolStripMenuItem->Size =
System::Drawing::Size(59, 24);
this->открытьФайлToolStripMenuItem->Text = L"Файл";
//
//  создатьToolStripMenuItem
//
this->создатьToolStripMenuItem->Name =
L"создатьToolStripMenuItem";
this->создатьToolStripMenuItem->Size =
System::Drawing::Size(244, 26);
this->создатьToolStripMenuItem->Text = L"Створити";

```

```

        this->создатьToolStripMenuItem->Click      +=      gcnew
System::EventHandler(this, &MyForm::создатьToolStripMenuItem_Click);
        //
        // открытьToolStripMenuItem
        //
        this->открытьToolStripMenuItem->Name      =
L"открытьToolStripMenuItem";
        this->открытьToolStripMenuItem->Size      =
System::Drawing::Size(244, 26);
        this->открытьToolStripMenuItem->Text = L"Відкрити";
        this->открытьToolStripMenuItem->Click      +=      gcnew
System::EventHandler(this, &MyForm::открытьToolStripMenuItem_Click);
        //
        // сохранитьToolStripMenuItem
        //
        this->сохранитьToolStripMenuItem->Name      =
L"сохранитьToolStripMenuItem";
        this->сохранитьToolStripMenuItem->Size      =
System::Drawing::Size(244, 26);
        this->сохранитьToolStripMenuItem->Text = L"Зберегти";
        this->сохранитьToolStripMenuItem->Click      +=      gcnew
System::EventHandler(this, &MyForm::сохранитьToolStripMenuItem_Click);
        //
        // сохранитьКакToolStripMenuItem
        //
        this->сохранитьКакToolStripMenuItem->Name      =
L"сохранитьКакToolStripMenuItem";
        this->сохранитьКакToolStripMenuItem->Size      =
System::Drawing::Size(244, 26);
        this->сохранитьКакToolStripMenuItem->Text = L"Зберегти як";
        this->сохранитьКакToolStripMenuItem->Click      +=      gcnew
System::EventHandler(this, &MyForm::сохранитьКакToolStripMenuItem_Click);
        //
        // резервнеКопіюванняToolStripMenuItem
        //

```

```

        this->резервнеКопіюванняToolStripMenuItem->Name =
L"резервнеКопіюванняToolStripMenuItem";
        this->резервнеКопіюванняToolStripMenuItem->Size =
System::Drawing::Size(244, 26);
        this->резервнеКопіюванняToolStripMenuItem->Text =
L"Резервне копіювання";
        this->резервнеКопіюванняToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MyForm::резервнеКопіюванняToolStripMenuItem_Click);
        //
        // формуватиЗвітToolStripMenuItem
        //
        this->формуватиЗвітToolStripMenuItem->Name =
L"формуватиЗвітToolStripMenuItem";
        this->формуватиЗвітToolStripMenuItem->Size =
System::Drawing::Size(244, 26);
        this->формуватиЗвітToolStripMenuItem->Text = L"Формувати
звіт";
        this->формуватиЗвітToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MyForm::формуватиЗвітToolStripMenuItem_Click);
        //
        // проЗавданняToolStripMenuItem
        //
        this->проЗавданняToolStripMenuItem->Name =
L"проЗавданняToolStripMenuItem";
        this->проЗавданняToolStripMenuItem->Size =
System::Drawing::Size(121, 24);
        this->проЗавданняToolStripMenuItem->Text = L"Про
завдання";
        this->проЗавданняToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MyForm::проЗавданняToolStripMenuItem_Click);
        //
        // проАвтораToolStripMenuItem
        //
        this->проАвтораToolStripMenuItem->Name =
L"проАвтораToolStripMenuItem";

```

```

        this->проАвтораToolStripMenuItem->Size =
System::Drawing::Size(104, 24);
        this->проАвтораToolStripMenuItem->Text = L"Про автора";
        this->проАвтораToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MyForm::проАвтораToolStripMenuItem_Click);
        //
        // відомістьПроФайлToolStripMenuItem
        //
        this->відомістьПроФайлToolStripMenuItem->Name =
L"відомістьПроФайлToolStripMenuItem";
        this->відомістьПроФайлToolStripMenuItem->Size =
System::Drawing::Size(159, 24);
        this->відомістьПроФайлToolStripMenuItem->Text =
L"Відомість про файл";
        this->відомістьПроФайлToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MyForm::відомістьПроФайлToolStripMenuItem_Click);
        //
        // openFileDialog1
        //
        this->openFileDialog1->FileName = L"openFileDialog1";
        //
        // button1
        //
        this->button1->Location = System::Drawing::Point(29, 327);
        this->button1->Name = L"button1";
        this->button1->Size = System::Drawing::Size(10, 29);
        this->button1->TabIndex = 5;
        this->button1->Text = L"Залишити ";
        this->button1->UseVisualStyleBackColor = true;
        this->button1->Visible = false;
        this->button1->Click += gcnew System::EventHandler(this,
&MyForm::button1_Click);
        //
        // button2
        //

```

```

this->button2->Location = System::Drawing::Point(455, 292);
this->button2->Name = L"button2";
this->button2->Size = System::Drawing::Size(377, 29);
this->button2->TabIndex = 6;
this->button2->Text = L"Видалити об'єкти вартістю менше
100грн.\r\n";

this->button2->UseVisualStyleBackColor = true;
this->button2->Click += gcnew System::EventHandler(this,
&MyForm::button2_Click);
//
// button3
//
this->button3->Location = System::Drawing::Point(0, 292);
this->button3->Name = L"button3";
this->button3->Size = System::Drawing::Size(295, 29);
this->button3->TabIndex = 7;
this->button3->Text = L"Сортувати за ціною\r\n";
this->button3->UseVisualStyleBackColor = true;
this->button3->Click += gcnew System::EventHandler(this,
&MyForm::button3_Click);
//
// button4
//
this->button4->Location = System::Drawing::Point(455, 327);
this->button4->Name = L"button4";
this->button4->Size = System::Drawing::Size(377, 29);
this->button4->TabIndex = 8;
this->button4->Text = L"Вибрати об'єкти за минулий
місяць\r\n";

this->button4->UseVisualStyleBackColor = true;
this->button4->Click += gcnew System::EventHandler(this,
&MyForm::button4_Click);
//
// MyForm
//

```

```

        this->AutoScaleDimensions = System::Drawing::SizeF(8, 16);
        this->AutoScaleMode
System::Windows::Forms::AutoScaleMode::Font;
        this->ClientSize = System::Drawing::Size(1005, 359);
        this->Controls->Add(this->button4);
        this->Controls->Add(this->button3);
        this->Controls->Add(this->button2);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->dataGridView1);
        this->Controls->Add(this->menuStrip1);
        this->Icon = (cli::safe_cast<System::Drawing::Icon^>(resources-
>GetObject(L"$this.Icon")));
        this->MainMenuStrip = this->menuStrip1;
        this->Name = L"MyForm";
        this->Text = L"MyForm";
        this->Load += gcnew System::EventHandler(this,
&MyForm::MyForm_Load);

        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))-
>EndInit();

        this->menuStrip1->ResumeLayout(false);
        this->menuStrip1->PerformLayout();
        this->ResumeLayout(false);
        this->PerformLayout();
    }
#pragma endregion
    String^ path="";
    bool SAVE = true;
    private: System::Void создатьToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
        if (SAVE || MessageBox::Show("Не збережені дані будуть втрачені,
продовжити?", "Попередження", MessageBoxButtons::YesNo, MessageBoxIcon::Warning)
== System::Windows::Forms::DialogResult::Yes)
        {
            dataGridView1->Rows->Clear();
            path = "";

```

```

    }
}

private: System::Void открытьToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    openFileDialog1->Filter = "Data file(*.DAT)|*.DAT";
    if (openFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK)
    {
        path = openFileDialog1->FileName;
        dataGridView1->Rows->Clear();
        BinaryReader^ file = gcnew BinaryReader(File::OpenRead(path));
        while (file->BaseStream->Position < file->BaseStream->Length && file-
>BaseStream->Length>6)
        {
            dataGridView1->Rows->Add(
                file->ReadInt16(),
                file->ReadString(),
                file->ReadString(),
                file->ReadDouble(),
                file->ReadInt16(),
                file->ReadString());
        }
        file->Close();
    }
}

private: System::Void сохранитьToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    if (path != "")
    {
        BinaryWriter^ file = gcnew BinaryWriter(File::Open(path,
FileMode::OpenOrCreate));
        for (int i = 0; i + 1 < dataGridView1->RowCount; i++)
        {
            file->Write(Convert::ToInt16(dataGridView1[0, i]->Value));
            file->Write(Convert::ToString(dataGridView1[1, i]->Value));
            file->Write(Convert::ToString(dataGridView1[2, i]->Value));
            file->Write(Convert::ToDouble(dataGridView1[3, i]->Value));
            file->Write(Convert::ToInt16(dataGridView1[4, i]->Value));
            file->Write(Convert::ToString(dataGridView1[5, i]->Value));
        }
    }
}

```

```

        }
        file->Close();
        SAVE = true;
    }
    else
        сохранитьКакToolStripMenuItem_Click(сохранитьКакToolStripMenuItem, e);
}

private: System::Void сохранитьКакToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e)
{
    saveFileDialog1->Filter = "Data file(*.DAT)|*.DAT|Text file(*.TXT)|*.TXT";
    if (saveFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK)
    {
        path = saveFileDialog1->FileName;
        BinaryWriter^ file = gcnew BinaryWriter(File::Open(path,
FileMode::OpenOrCreate));
        for (int i = 0; i + 1 < dataGridView1->RowCount; i++)
        {
            file->Write(Convert::ToInt16(dataGridView1[0, i]->Value));
            file->Write(Convert::ToString(dataGridView1[1, i]->Value));
            file->Write(Convert::ToString(dataGridView1[2, i]->Value));
            file->Write(Convert::ToDouble(dataGridView1[3, i]->Value));
            file->Write(Convert::ToInt16(dataGridView1[4, i]->Value));
            file->Write(Convert::ToString(dataGridView1[5, i]->Value));
        }
        file->Close();
        SAVE = true;
    }
}

private: System::Void dataGridView1_CellEnter(System::Object^ sender,
System::Windows::Forms::DataGridViewCellEventArgs^ e)
{
    SAVE = false;
}

private: System::Void button3_Click(System::Object^ sender, System::EventArgs^ e) {
    dataGridView1->Sort(dataGridView1->Columns[3],
ListSortDirection::Ascending);
}

```

```

private: System::Void button4_Click(System::Object^ sender, System::EventArgs^ e) {
    for (int i = 0; i + 1 < dataGridView1->RowCount; i++)
    {
        DateTime t1 = DateTime::Parse(Convert::ToString(dataGridView1[5, i]-
>Value));

        if ((t1.Month + 1 != DateTime::Now.Month || t1.Year !=
DateTime::Now.Year) && (t1.Month != 12 || t1.Year + 1 != DateTime::Now.Year ||
DateTime::Now.Month != 1))
        {
            dataGridView1->Rows->RemoveAt(i);
            --i;
        }
    }
}

private: System::Void формуватиЗвітToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e)
{
    String^ Zvit;
    saveFileDialog1->Filter = "Text file(*.TXT)|*.TXT";
    if (saveFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK)
    {
        Zvit = saveFileDialog1->FileName;
        StreamWriter^ file = File::CreateText(Zvit);
        for (int i = 0; i + 1 < dataGridView1->RowCount; i++)
        {
            file->WriteLine("Інвентарний номер:\t" + dataGridView1[0, i]-
>Value);

            file->WriteLine("Назва об'єкту:\t\t" + dataGridView1[1, i]-
>Value);

            file->WriteLine("ПІБ відповідальної особи:\t" +
dataGridView1[2, i]->Value);

            file->WriteLine("Вартість:\t\t" + dataGridView1[3, i]->Value);
            file->WriteLine("Кількість:\t" + dataGridView1[4, i]->Value);
            file->WriteLine("Дата постановки на облік:\t\t" +
dataGridView1[5, i]->Value);

            file->WriteLine("");
        }
        file->Close();
    }
}

```

```

    }
}

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    if (MessageBox::Show("Залишити у файлі лише данні з:\nФірма - TP-Link\nUSB - немає", "Підтвердження", MessageBoxButtons::YesNo, MessageBoxIcon::Question) == System::Windows::Forms::DialogResult::Yes)
    {
        for (int i = 0; i + 1 < dataGridView1->RowCount; i++)
            if ((Convert::ToString(dataGridView1[0, i]->Value) != "TP-Link") || Convert::ToBoolean(dataGridView1[5, i]->Value))
            {
                dataGridView1->Rows->RemoveAt(i); --i;
            }
    }
}

private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    if (MessageBox::Show("Залишити у файлі лише данні з:\nЦіною менше 100грн.", "Підтвердження", MessageBoxButtons::YesNo, MessageBoxIcon::Question) == System::Windows::Forms::DialogResult::Yes)
    {
        for (int i = 0; i + 1 < dataGridView1->RowCount; i++)
            if (Convert::ToInt16(dataGridView1[3, i]->Value) < 100)
            {
                dataGridView1->Rows->RemoveAt(i); --i;
            }
    }
}

private: System::Void проЗавданняToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    Task^ task = gcnew Task();
    task->ShowDialog();
}

private: System::Void проАвтораToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e) {
    Author^ a = gcnew Author();
    a->ShowDialog();
}

private: System::Void відомістьПроФайлToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^ e)

```

```

{
    if (!File::Exists(path))
    {
        MessageBox::Show("File not exists!");
        return;
    }

    FileInfo^ file = gcnew FileInfo(path);
    MessageBox::Show("Повне ім'я: " + file->FullName +
        "\nРозмір файла: " + (file->Length).ToString() + "
Байтів" +
        "\nЧас створення: " + file->CreationTime.ToString() +
        "\nЧас останнього редагування: " + file->LastWriteTime.ToString(),"Довідка");
}

private: System::Void MyForm_Load(System::Object^ sender, System::EventArgs^ e) {
    load^ Loading = gcnew load();
    Loading->ShowDialog();
}

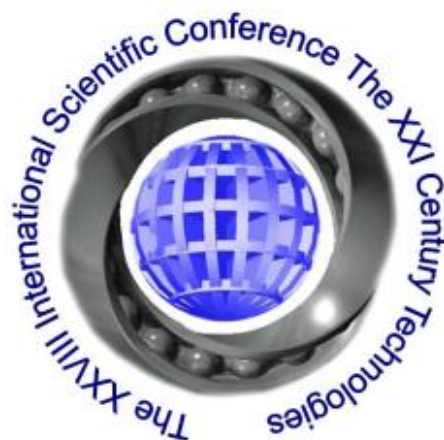
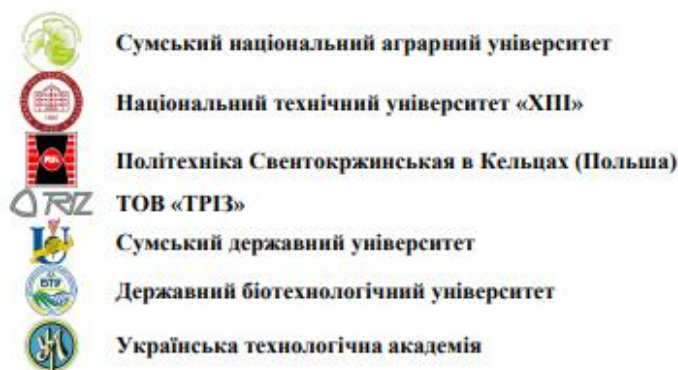
private:                                     System::Void
резервнеКопіюванняToolStripMenuItem_Click(System::Object^ sender, System::EventArgs^
e) {
    if (!Directory::Exists("Copy"))
        Directory::CreateDirectory("Copy");
    if (path != "")
    {
        FileInfo^ file = gcnew FileInfo(path);
        file->CopyTo("Copy\\" + file->Name, true);
    }
}

};

}

```

ДОДАТОК Г



ТЕХНОЛОГІЇ ХХІ СТОРІЧЧЯ

Збірник тез за матеріалами 28^{ої} міжнародної науково-практичної конференції
(23-25 листопада 2022 р.)

Частина 2

Секції: «Прогресивні технології в економіці»,
«Прогресивні технології в екології та енергоефективність»,
«Прогресивні технології у педагогіці»

Суми – 2022

Рисунок В.1 – Збірник тез 28 – ої міжнародної науково – практичної конференції

Джерело: [21]

ЗМІСТ

СЕКЦІЯ «ПРОГРЕСИВНІ ТЕХНОЛОГІЇ В ЕКОНОМІЦІ»

<i>Самойшкіна І.Д., доцент, СНАУ, м. Суми, Україна</i>	
МІЖНАРОДНИЙ ДОСВІД РОЗВИТКУ АГРАРНОГО СТРАХУВАННЯ В УКРАЇНІ	3
<i>Омельчук А.І., магістрант, Агаджанова С.В., к.т.н., доцент, Суми, СНАУ, Україна</i>	
ЗАКОНОДАВЧІ ЗАСАДИ ФУНКЦІОНУВАННЯ СИСТЕМИ НАДАННЯ СОЦІАЛЬНИХ ПОСЛУГ ДЛЯ СОЦІАЛЬНИХ ГРУП НАСЕЛЕННЯ	5
<i>Омельчук А.І., магістрант, Агаджанова С.В., к.т.н., доцент, Суми, СНАУ, Україна</i>	
ВДОСКОНАЛЕННЯ СИСТЕМИ НАДАННЯ СОЦІАЛЬНИХ ПОСЛУГ СПЕЦІАЛЬНИМ ГРУПАМ НАСЕЛЕННЯ ЗАСОБАМИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ	6
<i>Басанець Д.В., студент, СНАУ, Суми, Україна</i>	
ІНФОРМАЦІЙНА ПІДСИСТЕМА ОБЛІКУ КАДРОВОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ КОЛЕДЖУ ВСП «ГАФК СНАУ»	7
<i>Нечипоренко В.В., к.с.-г.н., доцент, СНАУ</i>	
СТРАТЕГІЧНІ НАПРЯМКИ СТАЛОГО РОЗВИТКУ АГРАРНОГО СЕКТОРУ ЕКОНОМІКИ	8
<i>Шакоцько А.М., студент, Толбатов А.В., к.т.н., доцент, СНАУ, Суми, Україна</i>	
ІНФОРМАЦІЙНА ПІДСИСТЕМА МАТЕРІАЛЬНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВСП «ГАФК СНАУ»	9
<i>Шакоцько С.С., молодший спеціаліст, Сумський національний аграрний університет</i>	
СУЧАСНІ НАПРЯМКИ СТРАХУВАННЯ КІБЕРРИЗИКІВ	11
<i>Білик Г.М., магістрант, Сумський національний аграрний університет, Україна</i>	
ОСОБЛИВОСТІ ЗЕМЛЕКОРИСТУВАННЯ У РОЗВИНЕНИХ КРАЇНАХ СВІТУ В ЧАСТИНІ ЕКСПЛУАТАЦІЇ СІЛЬСЬКОГОСПОДАРСЬКИХ УГІДЬ	12
<i>Грошовик Р.Ю., студент, Пасько Н.Б., доцент, Сумський НАУ</i>	
АВТОМАТИЗАЦІЯ ПОГОСПОДАРСЬКОГО ОБЛІКУ ТЕРИТОРІАЛЬНИХ ГРОМАД	15
<i>Кальченко Я., магістрант, СНАУ, Суми, Україна</i>	
ОЦІНКА ЕФЕКТИВНОСТІ ЗБУТОВОЇ ДІЯЛЬНОСТІ АГРАРНОГО ПІДПРИЄМСТВА	19
<i>Колеснікова Ю.С., магістрант, Сумський національний аграрний університет, Україна</i>	
ОСОБЛИВОСТІ ФОРМУВАННЯ МАРКЕТИНГОВОЇ СТРАТЕГІЇ ПЕРСОНАЛУ У ВЕТЕРИНАРНИХ КЛІНІКАХ	21
<i>Колозов Р.В., магістрант, Сумський національний аграрний університет, Україна</i>	
ПРОБЛЕМИ ПРОГНОЗУВАННЯ ВИКОРИСТАННЯ ЗЕМЕЛЬНИХ РЕСУРСІВ В УКРАЇНІ	24
<i>Колпаков В.В., магістрант, Сумський національний аграрний університет, Україна</i>	
МАРКЕТИНГ ВІДНОСИН ЯК ВАЖЛИВИЙ ЕЛЕМЕНТ КОНКУРЕНТНОЇ МАРКЕТИНГОВОЇ СТРАТЕГІЇ	26
<i>Лисенко К., магістрант, Сумський національний аграрний університет, Україна</i>	
МЕТОДОЛОГІЯ ТА АЛГОРИТМ ПРОВЕДЕННЯ ЗЕМЕЛЬНИХ ТОРГІВ	29
<i>Макарова В.В., д.е.н., доцент, Сумський національний аграрний університет, Україна</i>	
МЕТОДОЛОГІЧНІ ПІДХОДИ ДО ОЦІНКИ ЕФЕКТИВНОСТІ МАРКЕТИНГОВОЇ ДІЯЛЬНОСТІ В СОЦІАЛЬНИХ МЕРЕЖАХ	31
<i>Мозика А., магістрант, Сумський національний аграрний університет, Україна</i>	
ЕКОЛОГО-ЕКОНОМІЧНІ ТЕНДЕНЦІЇ ВИКОРИСТАННЯ СІЛЬСЬКОГОСПОДАРСЬКИХ ЗЕМЕЛЬ В УКРАЇНІ	34
<i>Селезень О.М., ст. викладач, Мотузний Ю.М., магістрант, СНАУ, Суми, Україна</i>	
СУТНІСТЬ ТА СУЧАСНІ МЕТОДИ ПРОСУВАННЯ ПРОДУКЦІЇ В СИСТЕМІ МАРКЕТИНГУ	36

Рисунок В.2 – Зміст 28 – ої міжнародної науково – практичної конференції

Джерело: [21]

УДК 658.5:004

Шакоотько А. М., студент, Толбатов А.В., к.т.н., доцент, СНАУ, Суми, Україна

ІНФОРМАЦІЙНА ПІДСИСТЕМА МАТЕРІАЛЬНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВСП «ГАФК СНАУ»

Матеріально-технічне забезпечення — це система організації обігу і використання праці підприємства, основних фондів і оборотних коштів. МТЗ також відповідає за їх розподіл між структурними сегментами та бізнес-одиницями та за споживання в процесі виробництва.

Основними завданнями матеріально-технічного забезпечення є:

1. Визначати потреби підприємства та структурного підрозділу в його матеріальних ресурсах;
2. Вивчати джерела забезпечення матеріальними ресурсами;
3. Організація закупівлі та доставки матеріальних ресурсів до місць споживання;

9

4. Розумне та економне використання матеріальних ресурсів;
5. Правильне зберігання матеріальних цінностей;
6. Оптимізація матеріально-технічного забезпечення;
7. Встановлювати економічні та обґрунтовані норми інвентаризації та контролювати їх умови;

До складу матеріальних ресурсів входять: комп'ютери, ноутбуки, навчальні матеріали, різні комплектуючі, технологічне устаткування і технологічне оснащення (пристрої, навчальні стенди та вимірювальні інструменти), нові транспортні засоби, обчислювальна техніка та інше устаткування, а також дрова, електроенергія, вода та ін..

Процес забезпечення підприємств матеріальними ресурсами називається матеріально-технічним постачанням. Цей процес характеризується певним набором специфічних функцій, робіт, операцій залежно від форм власності, системи розподілу та обміну продуктами матеріального виробництва.

Важливість матеріально-технологічного постачання підприємств дуже велика. Від правильної організації багато в чому залежить стан бізнесу. Тобто правильна організація матеріально-технічного забезпечення підприємства забезпечує нормальну роботу підприємства і сприяє загальному розвитку підприємства.

Діяльність підприємства складається з багатьох взаємопов'язаних процесів. Це означає, що ефективність і оптимальність кожного з них дуже важливі. Тому разом з виробничим, збутовим, маркетинговим та іншими процесами великий вплив на ефективність виробничого підприємства має матеріально-технічне забезпечення (МТЗ) підприємства.

Стратегія логістики (як функціональна стратегія) – це стратегія постачання, спрямована на керований розвиток і підвищення ресурсного потенціалу підприємства; існує відповідний план поділу підрозділів з метою створення конкурентоспроможного система виробництва та управління.

Для сучасної економіки характерні ринкові механізми здійснення матеріально-технічного постачання з тим чи іншим ступенем державного регулювання. Діапазон відносин може розпочатись від планової поставки матеріальних ресурсів за розпорядженням керуючого центру до вільного їх продажу на ринку з тим чи іншим ступенем використання ринкових механізмів. Таким чином, за допомогою матеріально-технічного постачання реалізуються відносини обміну засобів виробництва між суб'єктами господарської діяльності.

Але інформаційні системи економічних суб'єктів орієнтовані на створення та використання корпоративних розподілених процесів формування варіантів рішень, автоматизовану підтримку варіантів та прогноз діяльності суб'єктів, забезпечення виходу внутрішніх інформаційних потоків назовні.

Також може застосовуватись до матеріального забезпечення таке поняття як інвентаризація матеріальних засобів, яке є одним із радикальних способів перевірки та контролю за збереженням оборотних ресурсів і роботою закріплених за матеріальні цінностей відповідальних за цим осіб.

Рисунок В.3 Опублікована теза в 28 – ої міжнародної науково – практичної конференції

Джерело: [21]

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

МАТЕРІАЛИ

науково-практичної конференції
викладачів, аспірантів та студентів
Сумського НАУ

(25-28 квітня 2023 р.)

Рисунок В.4 – Матеріали науково – практичної конференції викладачів,
аспірантів та студентів

Джерело: [22]

Рудакова Д.О. РОЛЬ ЕКОЛОГІЧНОГО МАРКЕТИНГУ В ФОРМУВАННІ КОНКУРЕНТНИХ ПЕРЕВАГ ДЕРЖАВНОГО ПІДПРИЄМСТВА	339
Базарний С.О. УПРАВЛІННЯ ТОВАРНОЮ ПОЛІТИКОЮ НА ПІДПРИЄМСТВАХ МАЛОГО БІЗНЕСУ	340
Сергієнко В.Г. КАДРОВА БЕЗПЕКА ЯК ОСНОВНИЙ ЕЛЕМЕНТ ЕКОНОМІЧНОЇ БЕЗПЕКИ ПІДПРИЄМСТВА	341
Бельська А.О. РЕКЛАМНА ДІЯЛЬНІСТЬ НА ПІДПРИЄМСТВАХ РОЗДРІБНОЇ ТОРГІВЛІ ТА СПОСОБИ ЇЇ ВДОСКОНАЛЕННЯ	342
Дундук Д.С. РОЛЬ І ЗНАЧЕННЯ КРЕДИТУ В РОЗВИТКУ ЕКОНОМІКИ УКРАЇНИ.....	343
В'юненко О.Б. СУЧАСНІ ПРОБЛЕМИ ІНФОРМАЦІЙНОЇ КІБЕРГІГІЄНИ	344
Баталова А.Б. ГЕОІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ЯК ІНСТРУМЕНТ ЕКОЛОГІЧНИХ ДОСЛІДЖЕНЬ.....	345
Маяк А.О. ЦИФРОВІ ІНСТРУМЕНТИ ДЛЯ ПУБЛІКАЦІЇ ТА ПРОДАЖУ МУЗИКИ	346
Бусенко І.В. РОЗРОБКА ВІДЕОІГОР НА БАЗІ ТЕХНОЛОГІЇ P2E	347
Крамський А.С. ChatGPT – ТВІЙ МЕНТОР У НАВЧАННІ.....	348
Гервольський М.Д. АНАЛІТИЧНИЙ АНАЛІЗ РІЗНИХ ВИДИ ХОСТИНГІВ	349
Потапенко М.Т. АВТОМАТИЗОВАНА ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ СПОРТИВНОГО КОМПЛЕКСУ	350
Єременко О.В. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПЕРЕВІРКИ ПРАКТИЧНИХ РОБІТ ІЗ ВИБІРКИ ДАНИХ: ШЛЯХИ І ТЕХНОЛОГІЇ	351
Мотлюк М.В. А/В ТЕСТУВАННЯ ЯК МЕТОД МАРКЕТИНГОВИХ ДОСЛІДЖЕНЬ В ІТ	352
Олефіренко Д.В. ПРОГНОЗУВАННЯ КІЛЬКОСТІ ВЕЛИКОЇ РОГАТОЇ ХУДОБИ В УКРАЇНІ З ВИКОРИСТАННЯМ ЕКОНОМЕТРИЧНИХ МЕТОДІВ	353
Бабуров Д.В. ПРОГНОЗУВАННЯ КІЛЬКОСТІ СІЛЬСЬКОГОСПОДАРСЬКОЇ ПТИЦІ В УКРАЇНІ З ВИКОРИСТАННЯМ ЕКОНОМЕТРИЧНИХ МЕТОДІВ	354
Шакоотько А.М. ІНФОРМАЦІЙНА ПІДСИСТЕМА МАТЕРІАЛЬНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВСП «ГАФК СНАУ»	355
Басанець Д.В. ІНФОРМАЦІЙНА ПІДСИСТЕМА ОБЛІКУ КАДРОВОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ КОЛЕДЖУ ВСП «ГАФК СНАУ»	356
Хіманова А.О. ПРОБЛЕМИ УПРАВЛІННЯ ІТ ПРОЄКТАМИ СУЧАСНИХ НАВЧАЛЬНИХ СИСТЕМ	357
Сіденко А.С. ПРОБЛЕМИ ПОБУДОВИ САЙТІВ ДЛЯ СИСТЕМ ДИСТАНЦІЙНОГО НАВЧАННЯ ...	358
Шакоотько Є.С. СУЧАСНІ АЛГОРИТМИ АНАЛІЗУ ПРОДУКТИВНОСТІ ТА МОНІТОРИНГУ КОМП'ЮТЕРНИХ МЕРЕЖ	359
Грошовик Р.Ю. ОБГРУНТУВАННЯ НЕОБХІДНОСТІ АВТОМАТИЗАЦІЇ ВЕДЕННЯ ПОГОСПОДАРСЬКОГО ОБЛІКУ ТЕРИТОРІАЛЬНИХ ГРОМАД	360
Наконечний О.Б. АНАЛІЗ ПРОБЛЕМИ АВТОМАТИЗАЦІЇ РОЗРАХУНКУ БАЛАНСУ ГУМУСУ І ПОЖИВНИХ РЕЧОВИН В ҐРУНТІ ТА ПРОПОЗИЦІЇ ЩОДО ЇЇ ВИРІШЕННЯ	361
Мартинченко С.В. ОБРОБКА ВЕЛИКИХ ДАНИХ: ПРОБЛЕМИ І ШЛЯХИ ЇХ ВИРІШЕННЯ	362
Zhang Yuanyi THE KEY FACTORS IN PERSONAL DEVELOPMENT FROM THE PERSPECTIVE OF ENTERPRISES.....	363

Рисунок В.5 – Зміст матеріалів науково – практичної конференції

Джерело: Матеріали науково – практичної конференції

ІНФОРМАЦІЙНА ПІДСИСТЕМА МАТЕРІАЛЬНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВСП «ГАФК СНАУ»

Шакоцько А. М., студ. 4 (2ст) курсу ФЕІМ, спец. «Інформаційні системи та технології»
 Науковий керівник: к. т. н., доцент Толбатов А. В.
 Сумський НАУ

Матеріальне забезпечення підприємства - це комплексний процес, який включає у себе придбання, зберігання, використання та вивід з обігу різних матеріальних цінностей, необхідних для виробництва продукції та надання послуг. До складу матеріальних цінностей можуть входити різноманітні товари, обладнання, сировина, матеріали, паливо, запчастини, інструменти та інше.

Ефективне матеріальне забезпечення передбачає наявність системи управління запасами, яка дозволяє контролювати запаси та вчасно їх замовляти, оптимізувати рівень запасів, забезпечувати своєчасне замовлення та доставку матеріальних цінностей.

Крім того, ефективне матеріальне забезпечення передбачає наявність необхідних матеріальних ресурсів для виробництва продукції та надання послуг, забезпечення їх якісного зберігання та використання, контролювання витрат та зниження втрат матеріальних цінностей.

Інформаційні системи матеріального забезпечення підприємства - це комплекс програмних засобів, що дозволяють забезпечити ефективний управління потоками матеріальних ресурсів на підприємстві.

До складу таких систем можуть входити:

1. Системи управління запасами - дозволяють забезпечити точне відслідковування кількості і руху матеріальних ресурсів на складах та виробництві.

2. Системи моніторингу та контролю - дозволяють в режимі реального часу відстежувати переміщення матеріальних ресурсів на території підприємства.

3. Системи аналізу та прогнозування - аналізують дані про використання ресурсів та попередні замовлення для прогнозування майбутніх потреб підприємства в матеріальних ресурсах.

4. Системи автоматизації процесів - дозволяють автоматизувати процеси складського управління та операцій з попереднього замовлення ресурсів.

Важливість матеріально-технологічного постачання підприємств дуже велика. Від правильної організації багато в чому залежить стан бізнесу. Тобто правильна організація матеріально-технічного забезпечення підприємства забезпечує нормальну роботу підприємства і сприяє загальному розвитку підприємства.

Застосування інформаційних технологій матеріального забезпечення підприємства дозволяє знизити витрати на управління запасами, підвищити ефективність складських операцій, покращити контроль за рухом матеріальних цінностей та забезпечити оптимальний рівень запасів. В результаті цього підприємство може знизити свої витрати і підвищити свою прибутковість.

Також може застосовуватись до матеріального забезпечення таке поняття як інвентаризація матеріальних засобів, яке є одним із радикальних способів перевірки та контролю за збереженням оборотних ресурсів і роботою закріплених за матеріальні цінності відповідальних за цим осіб.

Врахування матеріальних цінностей є одним з найважливіших завдань адміністративно-господарської діяльності будь-якої організації. Спосіб обліку матеріальних запасів має визначити у своїй обліковій політиці кожна організація, оскільки без його вирішення неможлива організація складського обліку - неодмінного атрибуту системи обліку матеріальних цінностей на підприємстві.

Таким чином, можна зробити висновок, що наявність та правильна реалізація Інформаційної системи матеріального забезпечення підприємства є важливим фактором успішного управління підприємством, що забезпечує його стабільний розвиток та підвищення конкурентоспроможності.

Рисунок В.6 – Оpubлікована теза в матеріалах науково – практичної конференції

Джерело: [22]

ДОДАТОК Д

Інструкція користувача

Для запуску додатку потрібно запустити «exe» файл. Після цього завантажувється головна форма програми (рис. Д.1).

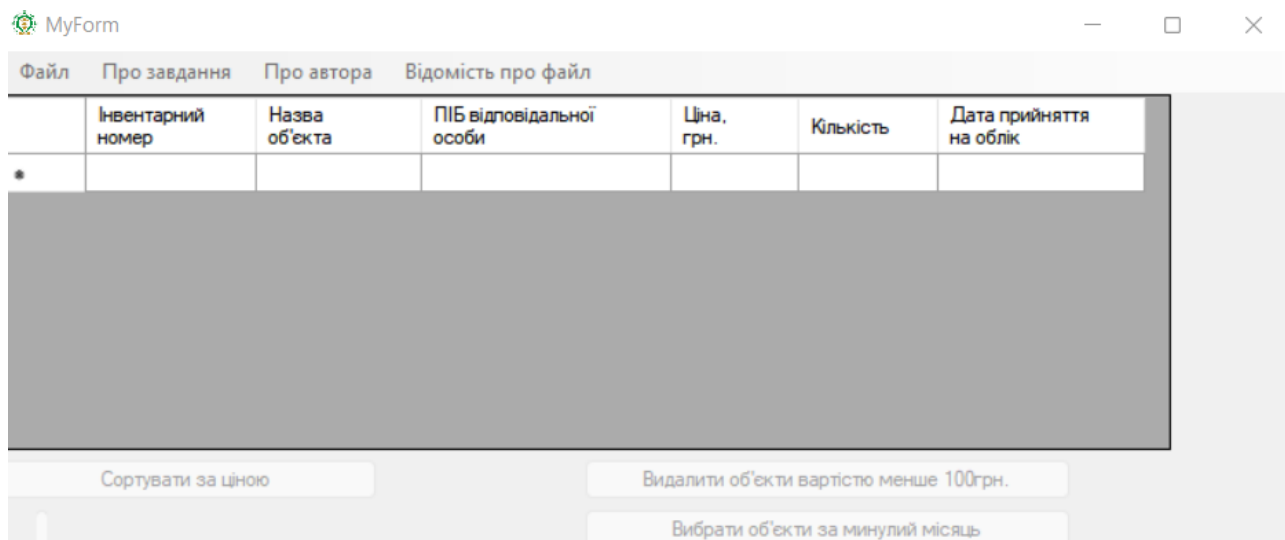


Рисунок Д.1 – Форма з заповненою таблицею за даними із файлу

Для завантаження даних необхідно вибрати в меню «Файл» вибрати бінарний файл і після вибору необхідного файлу відображається його вміст (рис.Д.2).

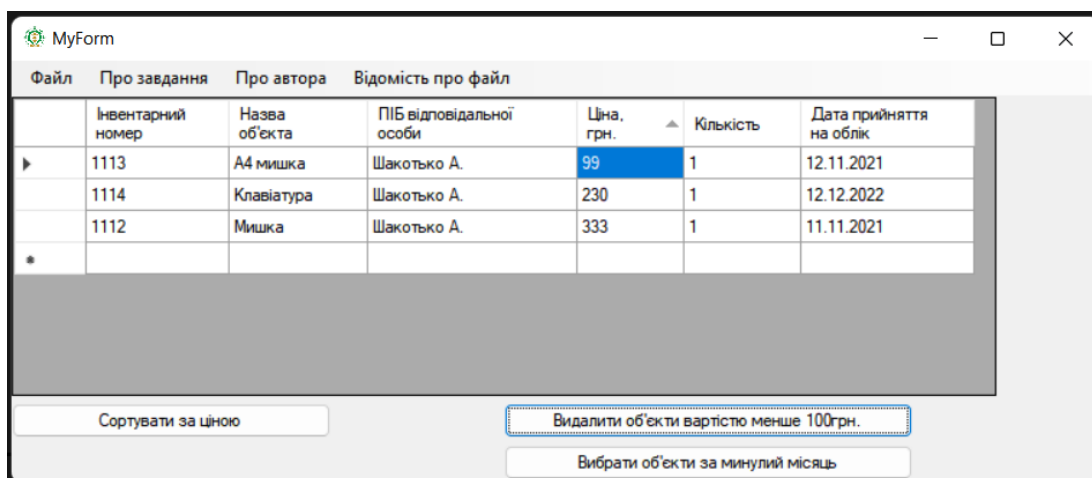


Рисунок Д.2 – Результат роботи вибраних об'єктів

Для сортування файлів можна вибрати з варіантів «Сортувати за ціною» та «Сортувати по назві об'єкта». Для видалення вибраних об'єктів потрібно натиснути на кнопку «Видалити об'єкти заданої вартості». Для того, щоб вибрати об'єкти прийняті на облік в заданий період натискаємо кнопку «Вибрати об'єкти за вибраний період».

У додатку дані зберігаються у бінарному виді, з розширенням файлів .DAT (рис. Д.3)

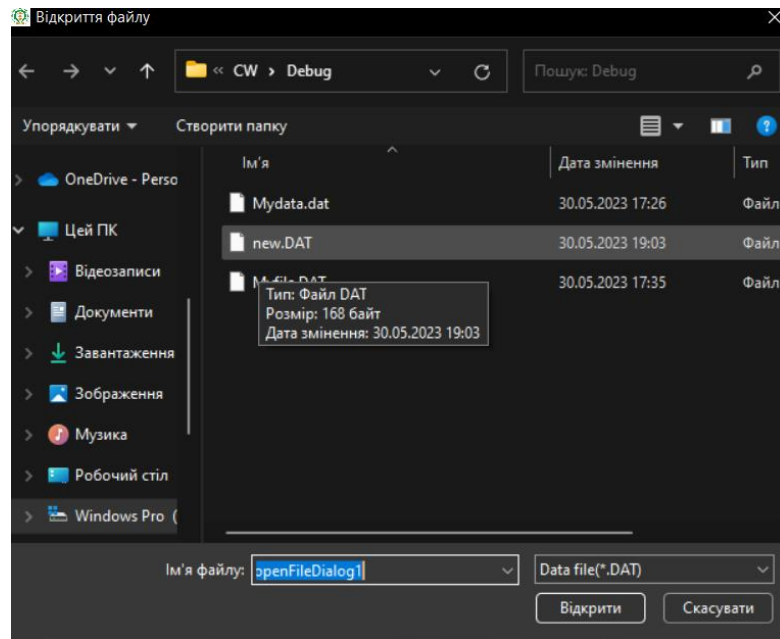


Рис Д.3 – Збережені дані у бінарних файлах

Описані рішення сприяють забезпеченню зручності та ефективності взаємодії з інформаційною системою. В результаті вдалих реалізаційних рішень було створено функціональний та ергономічний інтерфейс, що відповідає вимогам та потребам користувача.

Розроблений додаток зберігає звіти у форматі *.docx, так як в більшості випадків у користувачів буде присутня на комп'ютері програма для відкриття сформованих звітів (рис. Д.4).

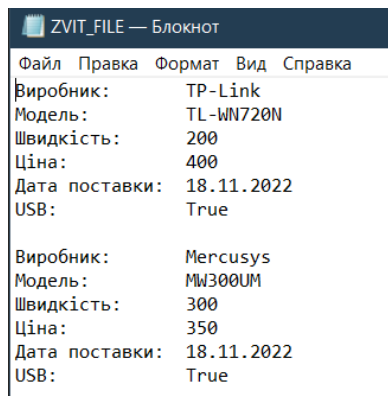


Рисунок Д.4 – Сформований звіт

Компоненти меню вдало інтегрувалися у інтерфейс інформаційної системи, забезпечуючи зручну та логічну організацію елементів керування. Розташування основних дій на видному місці і групування менш використовуваних дій у меню сприяє швидкому доступу та зменшує завантаження інтерфейсу.

Додавання стовпчиків у таблиці, відповідних параметрам записів у списку об'єктів, дозволяє зручно представляти та редагувати дані. Встановлення правильних назв та форматів запису стовпчиків сприяє зрозумілості та точності роботи з інформацією.

Реалізація роботи з бінарними файлами в рамках даного дипломного проекту також включає в себе важливий аспект - резервне копіювання інформації. Було розроблено та впроваджено механізм резервного копіювання, що дозволяє забезпечити захист даних в разі можливих випадків втрати або пошкодження файлів (рис. Д.5).

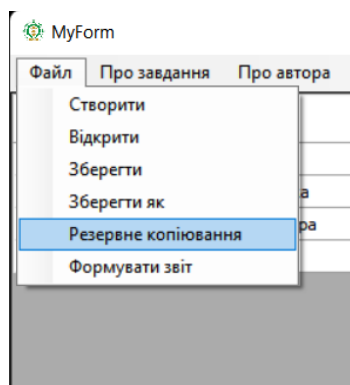


Рис. Д.5. – Резервне копіювання даних

Механізм резервного копіювання передбачає створення резервних копій бінарних файлів з використанням зручного та ефективного інтерфейсу. Користувач може вибрати, коли та як часто автоматично створювати резервні копії, а також здійснювати ручне створення копій у потрібний момент. Це дозволяє забезпечити збереження та захист даних, що є надзвичайно важливим аспектом в інформаційних системах.

Довідка про файл виводиться в окремому вікні (рис. Д.6).

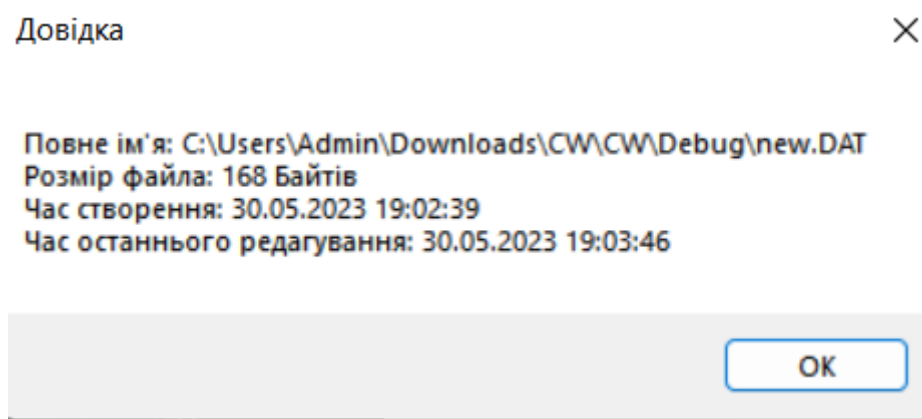


Рис. Д.6. – Довідка про файл

У програмі є виведення даних про автора, про завдання дипломного проєкта, та форма завантаження.

- 1) Про автора – містить короткі данні про автора проєкту
- 2) Про завдання – містить простий редактор для перегляду та редагування та збереження файлу із завданням.
- 3) Форма завантаження – форма, що з'являється при увімкненні програми, та поступово зникає, після чого запускається головна форма.

ДОДАТОК Е

Довідка про впровадження

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ
УНІВЕРСИТЕТ
Відокремлений структурний підрозділ
«Глухівський агротехнічний фаховий
коледж Сумського національного
аграрного університету»
вул. Терещенків, 36, м. Глухів, Сумська область
Україна, 41400, Тел./факс +38(05444) 23552
e-mail: hlukhiv_ksnau@ukr.net, ЄДРПОУ 44267623



MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
SUMY NATIONAL AGRARIAN
UNIVERSITY
Separate Structural Subdivision
“Hlukhiv Agrotechnical Professional
College of Sumy National
Agrarian University”
Tereshchenkiv Street, 36, Hlukhiv, Sumy Oblast
Ukraine, 41400, Tel./fax +38(05444) 23552
e-mail: hlukhiv_ksnau@ukr.net, ЄДРПОУ 44267623

Від 22.06.2023 № 40.1-17/60

На № _____ від _____

ДОВІДКА

про впровадження результатів
дипломної роботи бакалавра
Шакотька Артура Михайловича

Виконана студентом факультету економіки та управління 126 «Інформаційні системи та технології» Сумського національного аграрного університету Шакотько А.М. дипломна робота на тему "Інформаційна підсистема матеріального забезпечення інформаційної системи коледжу ВСП «ГАФК СНАУ»" та спроектований програмний додаток мають практичну значущість і розглянуті щодо впровадження у роботу економічного відділу ВСП «Глухівський агротехнічний фаховий коледж Сумського національного аграрного університету».

Директор ВСП «Глухівський
агротехнічний фаховий
коледж Сумського
національного аграрного
університету»



Анатолій ЛИТВИНЕНКО