

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор

_____ Олексій СМІРНОВ

“ _____ ” _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему

**“Дослідження та програмна реалізація системи логістики
оптимальних шляхів доставки товару”**

КББЗ-2023

Виконав здобувач вищої освіти
II курсу, групи КІ-23МЗ
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Лебеденко В.С.
« _____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук
_____ Кислун О.А.
« _____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань 12 “Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
Олексій СМІРНОВ
« 6 » вересня 2023 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Леbedенку Віталію Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи логістики оптимальних шляхів доставки товару

2. Керівник роботи Кислун Олег Андрійович, канд. техн. наук, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 36-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту 10.12.2023 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою роботи є дослідження та програмна реалізація системи логістики оптимальних шляхів доставки товару.

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна 1 аркуш

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

Показники економічної ефективності 1 аркуш

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	Завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	14.11.2023
Охорона праці	Оришака О.В.	06.10.2023	16.11.2023

7. Дата видачі завдання « 6 » вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2023 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Лебеденко В.С. Дослідження та програмна реалізація системи логістики оптимальних шляхів доставки товару. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

У даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення для сфери логістики доставки товару, яке призначено для зниження витрат на перевезення товару при наявній умові збереженості його якості.

Метою роботи є дослідження та програмна реалізація системи логістики оптимальних шляхів доставки товару.

Об'єктом дослідження є процес пошуку оптимальних шляхів доставки товару.

Предметом дослідження є методи пошуку оптимальних шляхів доставки товару.

Методи дослідження базуються на методах пошуку оптимальних шляхів, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи логістики оптимальних шляхів доставки товару.

У процесі роботи над програмною моделлю вдосконалено метод пошуку оптимальних шляхів доставки товару та розроблено вітчизняний продукт пошуку оптимальних шляхів доставки товару. У повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції щодо роботи з програмними засобами.

Програма призначена для виконання під управлінням операційної системи сімейства Windows.

Програму розроблено в середовищі Delphi 7.

Ключові слова: логістика, доставка товару, оптимальний шлях.

ABSTRACT

Lebedenko V.S. Research and software implementation of a logistics system for optimal delivery routes. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software was developed for the logistics of goods delivery, which is designed to reduce the cost of transporting goods under the existing condition of preserving its quality.

The purpose of the work is the research and software implementation of the logistics system of optimal ways of delivering goods.

The object of the study is the process of finding optimal ways of delivering goods.

The subject of research is methods of finding optimal ways of delivering goods.

Research methods are based on methods of finding optimal paths, methods of software development.

The result of the work is the software implementation of the logistics system of the optimal ways of delivering goods.

In the process of working on the software model, the method of finding optimal ways of delivering goods has been improved, and a domestic product for finding optimal ways of delivering goods has been developed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided. The program is intended for execution under the control of the operating system of the Windows family.

The program was developed in the Delphi 7 environment.

Keywords: logistics, delivery of goods, optimal way.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	4
ВСТУП.....	5
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	8
1.1 Призначення системи	8
1.2 Область застосування	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектури, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	19
2.3 Розгорнута постановка завдання	20
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	22
3.1 Опис функціонування системи	22
3.2 Розробка структурної схеми.....	31
3.3 Розробка функціональної схеми	34
3.4 Розробка діаграми процесів	36
4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ	39
4.1 Блок–схеми та опис алгоритмів функціонування систем	39
4.2 Захист розробленого програмного забезпечення.....	51

					ВКМР-123.23.0080.00.00.ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
<i>Розроб.</i>		<i>Лебеденко В.С.</i>			<i>Дослідження та програмна реалізація системи логістики оптимальних шляхів доставки товару</i>	Літ.	Арк.	Аркушів
<i>Перевір.</i>		<i>Кислун О.А.</i>				1	106	
<i>Н. Контр.</i>		<i>Коваленко А.С.</i>			ЦНТУ КІ-22МЗ			
<i>Затверд.</i>		<i>Смірнов О.А.</i>						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

А	–	Автентифікація
ЕП	–	Ескізний проект
ІТ	–	Інформаційні технології
ОС	–	Операційна система
ПЗ	–	Програмне забезпечення
ПК	–	Персональний комп'ютер
РП	–	Робочий проект
ТЗ	–	Технічне завдання
ТП	–	Технічний проект
BFS	–	Best-First Search
JPS	–	Jump Point Search
1PL	–	First Party Logistics
2PL	–	Second Party Logistics
3PL	–	Third Party Logistics
4PL	–	Fourth Party Logistics

										Лист
										4
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ					

ВСТУП

Актуальність теми. Незалежно від галузі сучасні фірми та підприємства зіштовхуються з питаннями логістики - транспортуванні товарно-матеріальних ресурсів при їх зберіганні та обробітку. Будь-які організації не можуть не працювати з постачальниками сировини та матеріалів з одного боку та споживачем продукції та послуг з іншого. Тож логістика існує в усіх видах господарської діяльності [1,2]. Логістика включає увесь комплекс операцій від постачальника сировини до покупця готового товару (аналіз ринку постачальників сировини, аналіз ринку споживачів, координація інтересів постачальників та споживачів, транспортування, зберігання, виробництво, розподіл тощо). На сьогодні на ринку логістичних послуг представлені логістичні провайдери з різним набором послуг та рівнем реалізації інтегрованого підходу до управління логістичними ланцюгами. Рівень 1PL (First Party Logistics) всі логістичні функції виконує одна компанія, яка є власником вантажів. Рівень 2PL (Second Party Logistics) - це спеціалізовані логістичні компанії, які почали виникати з 80-тих років, що надавали традиційний набір послуг з транспортування та управління складськими запасами. Рівень 3PL (Third Party Logistics) - на відміну від простих вантажоперевезень надаються не просто різного виду транспортні послуги, але й виконується увесь комплекс робіт, необхідний для просування товару по всьому ланцюжку поставок, включаючи експедирування, прийом, відвантаження і зберігання товарів на складах, страхування товарів, митне оформлення, перевалку вантажів тощо. Рівень 4PL (Fourth Party Logistics) - логістична інтегрованість, що має партнерський характер і полягає в управлінні логістичним ланцюгом клієнта, та системний логістичний інтегратор, основна функція якого не дистрибуція товарів, а

									Лист
									5
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

надання та координування інформаційних потоків. Рівень 5РБ провайдерів базується на сучасних мережевих комп'ютерних технологіях, як і в 4РБ відсутні права власності на матеріальні активи, здійснюється лише стратегічне управління ланцюгами постачання товарів [3,4].. До задач логістики входить дуже велике коло задач сучасного світу, які корінням входять у глибоке минуле, а за попередні роки накопичено величезний досвід в їх розв'язанні, та все ж задача про найкоротший шлях (пошук найкоротшого шляху, що загалом може мати різновиди: пошук найкоротших шляхів з одного входу, пошук найкоротших шляхів з одним виходом, пошук найкоротшого шляху між парою пунктів [5]) є базовою логістичною задачею, яка і в наш час не втратила актуальності, а задача пошуку оптимальних шляхів доставки товару може розглядатися як один з різновидів її прикладного спрямування .

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи логістики оптимальних шляхів доставки товару.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем логістики оптимальних шляхів доставки товару.
- Дослідження системи логістики оптимальних шляхів доставки товару.
- Програмна реалізація системи логістики оптимальних шляхів доставки товару.

Об'єктом дослідження є процес пошуку оптимальних шляхів доставки товару.

Предметом дослідження є методи пошуку оптимальних шляхів доставки товару.

										Лист
										6
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ					

Методи дослідження базуються на методах пошуку оптимальних шляхів, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі вирішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод пошуку оптимальних шляхів доставки товару.
- Розроблено вітчизняний продукт пошуку оптимальних шляхів доставки товару, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі логістики з пошуку оптимальних шляхів доставки товару.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведених у науковій літературі.

Робота апробована на:

- VI Міжнародній науково-практичній конференції "Інформаційна безпека та комп'ютерні технології", яка відбулася 20-21 квітня 2023 року у ЦНТУ м. Кропивницький;
- VII Міжнародної науково-практичної конференції до 30-ти річчя кафедри кібербезпеки та програмного забезпечення "Інформаційна безпека та комп'ютерні технології", яка відбулася 1 листопада 2023 року у ЦНТУ м. Кропивницький;

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи логістики оптимальних шляхів доставки товару, є актуальним завданням, яке потребує вирішення у даній магістерській роботі.

									Лист
									7
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Відповідно до Податкового кодексу України, під постачання товарів розуміють будь-яку передачу права на розпоряджання товарами як власник, у тому числі продаж, обмін чи дарування такого товару, а також постачання товарів за рішенням суду.

Під доставкою товару зазвичай розуміють транспортування (або перевезення) вантажів, як окрему господарську операцію, тобто по суті маємо процес переміщення вантажу з однієї точки в іншу з передачею доставленого вантажу одержувачу. Тож наявна задача управління матеріальними потоками при їхньому русі від первинного джерела до кінцевого споживача для отримання максимального зиску, який може досягатися шляхом зменшення витрат, які в свою чергу забезпечуються оптимізацією шляхів доставки, критерієм для якої є мінімізація.

Таким чином, основне завдання системи логістики оптимальних шляхів доставки товару є зниження витрат на їх перевезення (при наявній умові збереженості їх якості).

Загалом, з урахуванням введення понять:

- пункти (множина пунктів, можуть входити до маршрутів поставок);
- місце (пункт доставки - потрібне місце доставки товару, що потребується);
- потреба (необхідна кількості товару, що потребується);
- кількість (множина наявного товару - кількості товару в пунктах його розміщення);

									Лист
									8
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

самоврядування; промисловість; будівництво; торгівля; банківська сфера; медична сфера; туризм, а також безпосередньо і фізичні особи.

Таким чином, виходячи з вище перерахованого, дослідження та програмна реалізація застосування системи логістики оптимальних шляхів доставки товару є актуальною задачею, яка потребує вирішення у даній магістерській роботі.

КБПЗ – 2023

									Лист
									10
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектури, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Якщо задачі логістики розглядати крізь призму досягнення цілей основних оперативних процесів транспортування і зберігання вантажів, то їх можна поділити на три групи: глобальні, загальні, локальні. Головним глобальним завданням є забезпечення зростання прибутку за рахунок досягнення з найменшими витратами максимальної пристосованості фірм до мінливої ринкової ситуації, підвищення на ринку своєї частки та одержання переваг перед конкурентами. До загальних завдань логістики відносять створення інтегрованої ефективної системи регулювання й контролю за матеріальними та інформаційними потоками, які забезпечували б високу якість постачання продукції. Це створення інтегрованої системи регулювання матеріальних та інформаційних потоків; контроль за рухом матеріальних потоків; визначення стратегії та технології фізичного переміщення товарів; розробка засобів управління руху товарів; прогнозування обсягів виробництва перевезень, складування; вияв безбалансованості між потребами і можливостями закупівлі і виробництва; прогнозування попиту на товари, що виробляються чи переміщуються в рамках логістичної системи; розподіл транспортних засобів; організація передпродажного і післяпродажного обслуговування споживачів. До локальних задач логістики відноситься оптимізація виробничих запасів та максимальне скорочення часу зберігання та транспортування вантажів. [1].

									Лист
									11
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

Таким чином, якщо задачі логістики - це повсякденні задачі сучасного світу, які корінням входять в глибоке минуле, за попередні роки накопичено величезний досвід у їх розв'язанні. Широке використання обчислювальної техніки для пошуку оптимальних рішень привело до появи різноманітного програмного забезпечення в даній сфері, проте еволюційний розвиток потребує все нових програмних продуктів, тим більше, що час від часу виникають якісь нові, раніше не передбачені існуючими рішеннями, умови та специфічні вимоги. Отже, виникає задача аналізу набуття отриманого при розв'язанні логістичних задач, при цьому сам же огляд обмежено лише зацікавленістю в області оптимізації доставки.

Задачі доставки та їх рішення:

- транспортна задача (мінімізації витрат на розподілення одного товару або ресурсу з множини вузів базування до множини пунктів призначення відповідно до планів поставки);

- задача пошуку евклідового мінімального кістякового дерева (визначення на графі дерева з мінімальною сумою відстаней ребер, що поєднують всі існуючі вершини графа) в якості базового розв'язання використовуються алгоритми Прима та Крускала;

- задача комівояжера (знаходження мінімального шляху повного обходу вказаних пунктів з поверненням на початок маршруту при зазвичай наявній умові одноразового відвідування пунктів маршруту) загалом гарантовано точно може бути вирішена лише шляхом повного перебору всіх існуючих варіантів маршруту, проте в залежності від кількості пунктів маршруту наявна дуже велика складність перебору, а існуючі ж евристичні та комбіновані методи (найближчого сусіда, гілок та границь, ...) загалом знаходять непогані розв'язки, проте вони можуть бути гіршими за оптимальні;

									Лист
									12
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

- задача про найкоротший шлях (пошук найкоротшого шляху, що загалом може мати різновиди: пошук найкоротших шляхів з одного входу, пошук найкоротших шляхів з одним виходом, пошук найкоротшого шляху між парою пунктів) - базова задача, яку і покладено в основу розробки алгоритмів оптимізації маршрутизації (алгоритми: Дейкстри, Беллмана-Форда, пошуку A^* , Флойда-Воршелла, Джонсона, ...).

Знаходження найкоротших шляхів з одного входу до всіх інших пунктів алгоритм Дейкстри проводить як покрокову мінімізацію шляху від пункту при почерговому оціненні всіх можливих переміщень з пункту та визначенні з фіксацією оптимальнішого переміщення (початкові шляхи до пунктів до визначаються потенційно більшими за можливі, початковий шлях з входу визначається нулем, переміщення оптимізаційного пошуку з пункту проводиться по напрямку довжини шляху, що відшукується, з виключенням відпрацьованого пункту, завершення пошуку шляху визначається відсутністю пунктів для пошуку нового оптимальнішого кроку шляху).

Знаходження найкоротшого шляху з одного входу до всіх інших пунктів алгоритм Беллмана-Форда проводить як покрокову мінімізацію шляху до пункту при почерговому оціненні всіх можливих переміщень з кожного з пунктів та визначенні з фіксацією оптимальнішого (початкові шляхи до пунктів визначаються потенційно більшими за можливі, початковий шлях з входу визначається нулем, завершення пошуку шляху визначається відсутністю оптимальнішого шляху до жодного з множини пунктів).

Алгоритм пошуку A^* - це варіант алгоритму Дейкстри з евристичною оцінкою відстані між вузлом і фінішем для надання переваги вибору відкриття вузлів з метою усунення аналізу потенційно довгих шляхів.

Алгоритм Флойда-Воршелла (інші назви: Флойда, Роя-Воршелла, Роя-Флойда или WF) використовується для знаходження оптимального шляху

між парою (двома пунктами), що побудовано на покроковій мінімізації шляху між парою через третю точку (завершення алгоритму пошуку оптимального шляху виявляється відсутністю мінімізаційного шляху між будь-якою парою).

Алгоритм Флойда-Воршелла Джонсона призначено для знаходження найкоротших шляхів між усіма парами вершин та базується на зміні ваги та використанні алгоритмів Беллмана-Форда та Дейкстри. [5, 6].

ANT-Logistics - Український хмарний сервіс для планування маршрутів доставки з урахуванням пробок, тимчасових вікон, завантаження, вартості транспорту та інших параметрів [7, 8, 9].

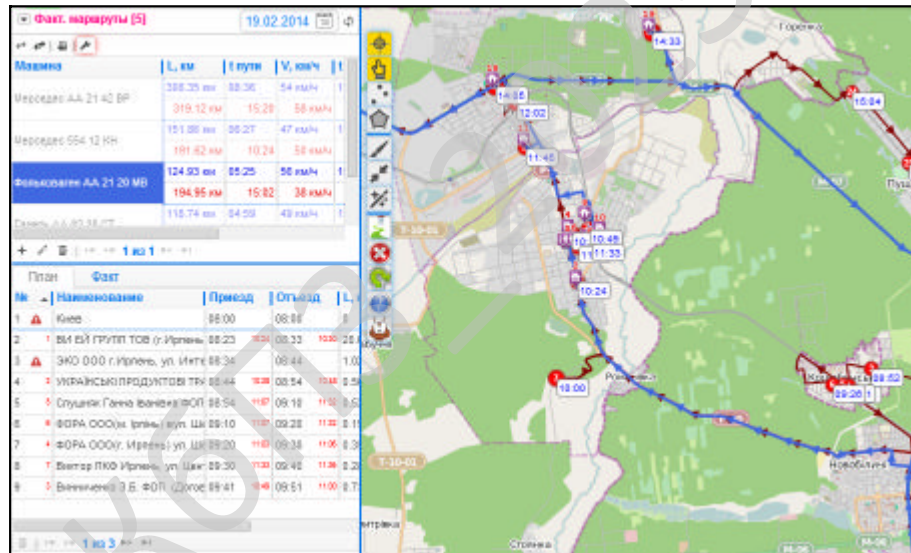


Рисунок 2.1 - Інтерфейс ANT-Logistics

Сервіс здійснює контроль за виконанням маршрутів, використовуючи GPS-моніторинг або авторський мобільний додаток для кур'єрів. До системи підключено понад 600 клієнтів з України та інших країн Європи. ANT-Logistics автоматично розподіляє замовлення між автомобілями і виконує розрахунок маршрутів, враховуючи більше 60 параметрів. Це дозволяє ще до

обмежень скорочуючи витрати. За допомогою програми вдається знизити вплив людського фактору, підвищити якість доставки та надати повний та оперативний контроль над автопарком. Завдяки АВМ Rinkai TMS один логіст може ефективно управляти до 10 складів. Програма має зручний web-інтерфейс з підтримкою української мови та мобільний додаток на платформі Android для водіїв, що дозволяє отримувати інформацію про маршрути доставки в режимі реального часу. Система надає керівникам логістики, власникам компаній, логістам широкі можливості для реалізації комплексної стратегії зниження транспортних витрат [10, 11]..

Можливості системи: Оптимальне формування маршрутів доставки; Складання маршрутів з урахуванням усіх обмежень: кілометраж, вартість доставки, часові вікна, вагогабаритні параметри, пасивний час перебування на складі; Розрахунок вартості доставки одиниці товару; Комплексний аналіз ключових параметрів використання та управління транспорту; Порівняння плану та фактичний результат виконання маршрутів; Функціональний мобільний додаток на платформі Android для водіїв; Інтеграція з обліковою системою; Раціоналізація у використанні найманого автопарку; Моделювання фінансового результату в залежності від вибору транспорту [11].

TMS UIS - система управління кур'єрською доставкою.

Комплексне рішення для автоматизації служби доставки, що покриває весь цикл - від моменту забору і до "останньої милі"

Функціонал системи TMS UIS: Обробка забору вантажів; Первинна обробка вантажів на складі; Сортування вантажів по кур'єрам; Планування доставки по зонам; Планування доставки по тимчасовим вікнам; Контроль доставки вантажів; Контроль залишків у кур'єра; Обробка повернень; Обробка каси кур'єра і післяплати; Моніторинг завантаженості кур'єрів; Підтримка роботи з РРО [12].

									Лист
									16
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

Впровадження TMS UIS здійснюється при наявності штрафів за зрив термінів доставки, при необхідності зниження витрат на логістику, коли потрібен контроль персоналу і руху товару, коли необхідно усунути простої, коли необхідно підвищити продуктивність кур'єрів, коли потрібно підвищити лояльність кінцевих клієнтів.

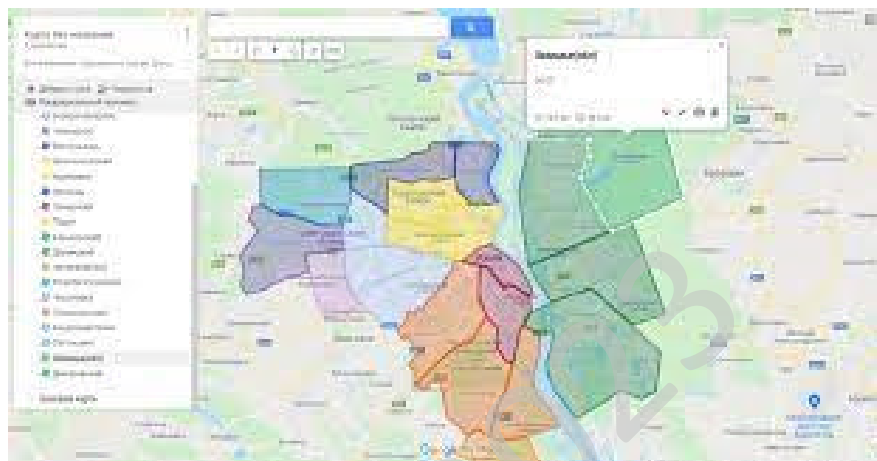


Рисунок 2.3 - Інтерфейс TMS UIS

Переваги TMS UIS: Повний цикл – від відправки до доставки; Зменшення простою до 0; Підвищення продуктивності кур'єрів до + 30%; Доставка по часовим вікнам; Контроль залишків у кур'єра і на рампі в реальному часі; Можливість роботи з безліччю міст і складів обслуговування; Підтримка часткової видачі замовлень; Система побудована на Oracle; Найвища відмовостійкість; Інтеграція з Google Maps; Підтримка роботи з РРО; Вбудований API [12].

Загалом ринок програмних продуктів України у сфері логістики характеризується досить значним різноманіттям представлених рішень, і на 2023 рік надамо перелік інших розробок, скоротивши їх опис:

- **Relog** - хмарний сервіс для внутрішньоміської маршрутизації, що дозволяє доставляти по геозонах з урахуванням тимчасових вікон та план-факт аналізом;

- **Logist.UA** - TMS система управління транспортом підприємства (Автоматизація управління доставками, планування маршрутів та моніторингу пересування автотранспорту підприємства);

- **EdiCourier** – рішення для автоматизації служб доставки та виїзного обслуговування клієнтів;

- **B2Field** – рішення для управління службою доставки: від планування маршрутів до збору клієнтських відгуків;

- **4logist** - багатофункціональний сервіс для транспортних та логістичних компаній, який дозволяє оптимізувати та спростити роботу всіх служб, пов'язаних з організацією вантажоперевезень;

- **Foodpicasso** - мобільний додаток для служби доставки їжі, ресторанів та кафе;

- **SERVIO Delivery** - модульний програмний продукт для швидкої та зручної роботи операторів, організації бізнес-процесів та їх автоматизації [13].

Оскільки кінцевою метою дослідження в рамках, якого наводиться огляд, є розробка програмного забезпечення для пошуку оптимальних маршрутів за певних специфічних умов постачання, то, відповідно, саме алгоритми пошуку найкоротшого шляху лягають в основу подальшої розробки та потребують адаптації, що враховуватиме додаткову затратність накладених умов.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Одним із робочих інструментів програміста безпосередньо як засобу є середовище програмування. Для цього на даному етапі розвитку ІТ сфери застосовуються візуальні середовища програмування.

Візуальне середовище програмування - інтегроване середовище розробки програмних засобів (IDE, яке містить редактор вихідного коду, компілятор чи/або інтерпретатор, засоби автоматизації збірки та засоби для спрощення розробки графічного інтерфейсу користувача). Середовища для візуального програмування також надають змогу конструювати програми шляхом оперування графічними об'єктами [14].

Delphi7 - можливо й не є найсучаснішим середовищем візуального програмування, проте - це надійне, доступне, поширене й таке, що відповідає всім потребам проектування додатків ОС Windows : Система програмування в Delphi базується на використанні візуальної технології програмування RAD (Rapid Application Development – середовище швидкої розробки додатків), в основу якої покладено використання стандартних візуальних компонентів - бібліотечних стандартизованих класів, що значно скорочує час на розробку програм [15,16]

В порівнянні Delphi з іншим популярним і можливим альтернативним інструментом програмування C++ зазначимо, що:

- структури програм на Delphi та C++ загалом схожі, оскільки це продукти однієї компанії Borland (хоч програма Delphi поділяється на заголовну частину, розділ описів і тіло програми; а у C++ ж не має чіткого поділу на заголовну частину та розділ описів, змінні можна оголошувати, на відміну від Delphi, прямо в тілі програми);

- типи даних C++ і Delphi схожі, переважно залежать від версії компілятора;
- обидві мови підтримують великі обсяги даних;
- обидві мови є об'єктно-орієнтованими, що надають порівняні фундаментальні можливості (об'єктно-орієнтований підхід, робота з великими обсягами даних, можливість низькорівневого програмування), то до них застосовують однакові технології створення програм: структурне програмування, об'єктно-орієнтоване програмування.

Delphi призначений для професійних розробників, що бажають дуже швидко розробляти додатки в архітектурі клієнт-сервер. Delphi робить невеликі за розмірами високоефективні виконувані модулі (.exe і .dll).

Система програмування Delphi розрахована на програмування різних застосувань і надає велику кількість компонентів для цього. До того ж працедавців цікавить, передусім, швидкість і якість створення програм, а ці характеристики можуть забезпечити тільки середовище візуального проектування, здатне взяти на себе значні об'єми рутинної роботи з підготовки додатків, а також погоджувати діяльність групи постановників, кодувальників, тестерів і технічних письменників. Можливості Delphi повністю відповідають подібним вимогам і підходять для створення систем будь-якої складності [15,16]

Для побудови системи логістики оптимальних шляхів доставки товару, базуючись на наявності в якості засобу розробки, обрано Delphi 7.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи логістики оптимальних шляхів

										Лист
										20
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ					

доставки товару.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методика побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформувані висновки про виконаний обсяг робіт та одержані результати.

										Лист
										21
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ					

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Огляд задачі знаходження найкоротшого шляху

Задача про знаходження найкоротшого шляху між парою вершин в теорії графів полягає в відшукуванні такого ланцюга між цією парою вершинами, коли б сума ваг ребер, з яких він складається, була мінімальна [17].

Постановка задачі знаходження найкоротшого шляху.

Постановка класичної задачі про найкоротший шлях. Нехай на визначеному наведеному графі G зі скінченим набором вершин V та ребер E із зазначеною функцією ваги f , для пари заданих вершин v^a та v^k , необхідно знайти найменший шлях p (ланцюг від v^a до v^k) серед усіх можливих шляхів P , що поєднують v^a та v^k з урахуванням направленості [17, 18, 19]

Постановка класичної задачі дещо відрізняється від наступних узагальнень:

- задача про знаходження найкоротших шляхів з одного входу v^a - це пошук найкоротших шляхів від вхідної вершини v^a до всіх інших вершин графу;

- задача про знаходження найкоротших шляхів з одним виходом v^k - це пошук найкоротших шляхів з усіх вершин графу до однієї вихідної вершини v^k ;

- задача про знаходження найкоротших шляхів для всіх пар - це пошук найкоротших шляхів між кожною існуючою парою вершин графа [17].

									Лист
									22
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

Алгоритми пошуку найкоротшого шляху..

На даний час існує багато алгоритмів для розв'язання задачі пошуку шляхів. До більш поширених слід віднести: алгоритм пошуку в глибину, пошуку у ширину, алгоритм Дейкстри, алгоритм Беллмана-Форда, алгоритм A^* , алгоритм Флойда-Воршела, алгоритми D^* , алгоритми BFS, алгоритми JPS, ... [18,19,20,21,22,23,24].

Алгоритм пошуку в глибину (Depth-first search, DFS) - один з алгоритмів обходу графа. Згідно до алгоритму:

- у графі G вибирається довільна вершина a , і з неї починається пошук, після цього вважається пройденою;

- наступний крок – вибирається ребро (a,b) , інцидентне вершині a (з урахуванням напрямку) та переходимо до вершини b (ребро (a,b) після переходу вважається переглянутим і називається ребром дерева, а вершина a називається батьківською по відношенню до вершини b);

- у загальному випадку, коли ми перейшли в будь-яку вершину графа, у випадку вершини b , виникають дві альтернативи подальшого виконання алгоритму:

- якщо всі ребра, інцидентні вершині b - вже переглянуті, то повертаємося до її батьківської вершини (для вершини b це вершина a), і продовжуємо пошук з цієї вершини (вершина b з цього моменту називатиметься повністю сканованою);

- якщо існують не переглянуті ребра, інцидентні b , то вибираємо одне з таких ребер (ребро з цього моменту вважатиметься переглянутим), при цьому розглядаються два випадки:

- якщо нова вершина раніше не була пройдена, то, використовуючи знайдене ребро, проходить перехід до нової вершини і продовжуємо пошук з неї (саме ребро переходу називається ребром дерева, а його початкова вершина - батьком знайденої вершини);

- якщо нова вершина раніше була пройдена, то продовжуємо пошук іншого не пройденого ребра (в цьому випадку наткнулись на так зване зворотне ребро);

- обхід графа в глибину завершується, коли ми повертаємося в корінь і всі вершини графа пройдені;

- якщо після повернення в корінь залишаються не пройдені вершини, можна вибрати одну з них і повторювати процес, і робити так до тих пір, поки не пройдемо всі вершини графа.

Алгоритм пошуку в глибину може обмежено використовуватись як алгоритм пошуку шляху. Його призначення - це дослідження топологічних властивостей [20,21].

Алгоритм пошуку в ширину (Breadth-first search, BFS) - один з алгоритмів обходу графа дозволяє знайти шляхи з однієї вершини орієнтованого або неорієнтованого графа до всіх інших вершин. що містить найменшу кількість ребер. Згідно до алгоритму пошуку в ширину із заданої стартової вершини:

- всі вершини позначаються як не пройдені;

- стартова вершина позначається активною вершиною (при будь-якій подальшій активізації не відвіданих вершини, для них задається глибина пошуку в ширину - ціле число, що вказує кількість кроків до вершини від початкової);

- на наступному кроці досліджуються ребра, інцидентні вершині, що стала активною, у загальному випадку, можливі два варіанти подальших дій:

- якщо всі ребра, інцидентні вершині, переглянуті, вона перестає бути активною і перетворюється в повністю скановану вершину. В такому випадку вибирається наступна активна вершина з найменшою глибиною пошуку в ширину, і дія повторюються;

Згідно до алгоритму:

- встановлюється відстань до стартової вершини рівною 0, а до решти вершин - нескінченності.

- виконується $n-1$ ітерація (де n – кількість вершин у графі); на кожній ітерації алгоритм розглядає всі ребра графа та оновлює відстані до кожної вершини, якщо знайдено коротший шлях через поточну вершину.

- оновлення відстаней продовжується до досягнення оптимального рішення (відсутність оновлення) або поки не буде виконано $n-1$ ітерація (у випадку, коли після $n-1$ ітерації відбувається оновлення відстаней, визначається наявність негативного циклу в графі).

Після завершення алгоритму можна отримати найкоротші шляхи до всіх вершин графа, а також визначити наявність негативних циклів [18,20,21]..

Алгоритм Флойда-Воршела - це алгоритм для знаходження найкоротшої відстані між всіма вершинами графа, в якому можуть бути наявні з вагою ребер від'ємного значення.

Згідно до алгоритму:

- вершини графа представляються як пронумеровані від 1 до n ;

- алгоритм використовує для розрахунку матрицю суміжності розмірності $n \times n$, що містить поточні значення довжин між вершинами, спочатку заносяться наявні довжини дуг (відомі довжини шляхів між вершинами, всі інші включно з діагональними елементами приймаються нескінченними);

- на кожному кроці алгоритму, для кожної пари вершин визначається можливість зменшення поточної відстані через проміжну вершину, якщо таке покращення можливе, то оновлюється поточне значення відстані шляхом зміни відповідного значення матриці суміжності;

- виконання алгоритму закінчується при відсутності покращень шляху .

					ВКРМ-123.23.0080.00.00.ПЗ	Лист
Змн.	Лист	№ докум.	Підпис	Дата		26

Алгоритм Флойда-Воршела виконується за n^3 ітерацій і не може бути ефективним для графів із великою кількістю вершин [18,20,21].

Алгоритм A^* («A star») - це евристичний алгоритм пошуку найкоротшого шляху між двома вершинами графу з додатними вагами ребер. Алгоритм A^* подібний до алгоритму Дейкстри, у якому для визначення порядку обходу вершин теж використовується черга з пріоритетом. Проте, на відміну від алгоритму Дейкстри, у алгоритмі A^* використовується пріоритет кожної неопрацьованої вершини, що враховує не лише поточну довжину шляху від стартової точки, а й додатково допоміжну функцію (евристику), аби скеровувати напрям пошуку до кінцевої точки.

В алгоритмі A^* пріоритет можна визначити за допомогою функції:

$$f(x) = g(x) + h(x),$$

де $g(x)$ - функція вартості шляху від початкової вершини до x ,

$h(x)$, - евристична функція, яка оцінює вартість шляху від вершини x до кінцевої [18,20,21].

Згідно до алгоритму:

- для вершин, на яких шукається оптимальний шлях, вводиться поняття прохідності («відкрита» - ще не пройдена, але до неї визначено та обраховано шлях, «закрита» - вже пройдена);

- обирається стартова вершина, як поточна;

- сусідні вершини до поточної, якщо такі існують, відкривається (перевідкривається, якщо оптимізується їх шлях) і визначається та обраховується їх шлях, а поточна вершина позначається як «закрита»;

- з відкритих вершин, якщо такі існують та якщо виконується умова можливості знаходження більш оптимального шляху, за значенням $f(x)$ обирається наступна поточна вершина й повторюється відкриття (перевідкриття) сусідніх вершин, інакше алгоритм завершується: «з знаходженням шляху» - коли кінцева вершина досягнута і «з відсутністю

шляху» - коли кінцеву вершину не досягнуто, а відкриті вершини закінчилися.

Алгоритм A^* фактично удосконаленням алгоритму Дейкстри [21].

Крім зазначених алгоритмів пошуку оптимальних шляхів, існують й інші: алгоритм D^* (сімейство з трьох алгоритмів інкриментного пошуку), алгоритм Best-First Search (сімейство алгоритмів BFS пошуку за першим кращим - алгоритм A^* може слугувати як приклад таких алгоритмів), алгоритми Jump Point Search (алгоритми JPS з перестрибуючими точками - є покращеннями алгоритму пошуку шляху A^* , в якому наявна велика кількість вершин, що повинні бути переглянуті, але була пропущена, що дозволяє прискорити алгоритм пошуку шляху, розглядаючи тільки частину вершин, як таку, що ввійде до оптимального шляху).

Оскільки кінцевою метою є розробка програмного забезпечення для пошуку оптимальних маршрутів за певних специфічних умов постачання, то, відповідно, саме класичні алгоритми пошуку найкоротшого шляху лягають в основу побудови алгоритму пошуку.

Вивчивши роботу класичних алгоритмів пошуку A^* , Дейкстри, Беллмана-Форда та інших, скорегуємо алгоритм пошуку маршрутів оптимального постачання [18,19,20,21].

Формально маємо потребу в адаптації алгоритму класичної задачі про найкоротші шляхи з одним виходом, на яку накладено умову затратності використання додаткового прохідного пункту в маршруті. Оскільки вартість вантажних робіт загалом є основною складовою витрат, то для врахування вказаних затрат пропонується ввести в якості опису матрицю затрат. Нехтуючи можливою відмінністю величин вантажних робіт в одному прохідному пункті для різних вхідних та вихідних шляхів, а залишивши лише відмінність вантажних робіт в самому пункті - матриця затрат буде представлена лише вектором.

- дані про затрати, що пов'язані з використанням пункту, як проміжного;
- дані про кінцевий пункт маршруту доставки товару;
- дані про потребу товару в пункті доставки.

3.2 Розробка структурної схеми

Розгляд структури програми можливий з двох позицій: розгляд, як структурних одиниць програми самої Delphi або як описання зв'язків між різними структурними одиницями програми (структурні одиниці програми розглядаються в якості окремих частини - блоків, що виконують різні функції) та проходженням інформаційних потоків через них. Будь-яка програма Delphi складається з файла проекту (файл з розширенням DPR) і одного чи кількох модулів (файли з розширенням PAS). Структурно модулі - це окремі програмні одиниці, що реалізують окремі частини програми. [25]. Тож для побудови програми скористаємося можливістю мінімального використання модулів - один модуль. Розглянемо структурну програму через призму виконання її з боку функції. Структурна схема системи зображена на рисунку 3.1.

Зі схеми можна побачити, що до програми поступають наступні інформаційні потоки:

- пункти - кількість пунктів;;
- затрати в пункті, використаному як прохідний - матриця затрат в пункт;
- наявний товар - множина (вектор) наявного товару в пунктах;
- маршрути - граф, у якому наявний маршрут відмінний від нуля (загалом затрати за маршрутом поєднано в цей же масив, що ним описано граф маршрутів);
- затрати за маршрутами - матриця затрат на доставку з пункту в пункт (як вже зазначалося при опису вхідного потоку, яким визначалися наявні маршрути

використовується один спільний масив для опису - тобто можливе об'єднання вхідних потоків маршруту та затрати за маршрутами):

- пункт доставки - номер (порядковий номер кінцевого маршруту доставки);
- потреба - величина, що вказує потрібний об'єм товару в пункті доставки;

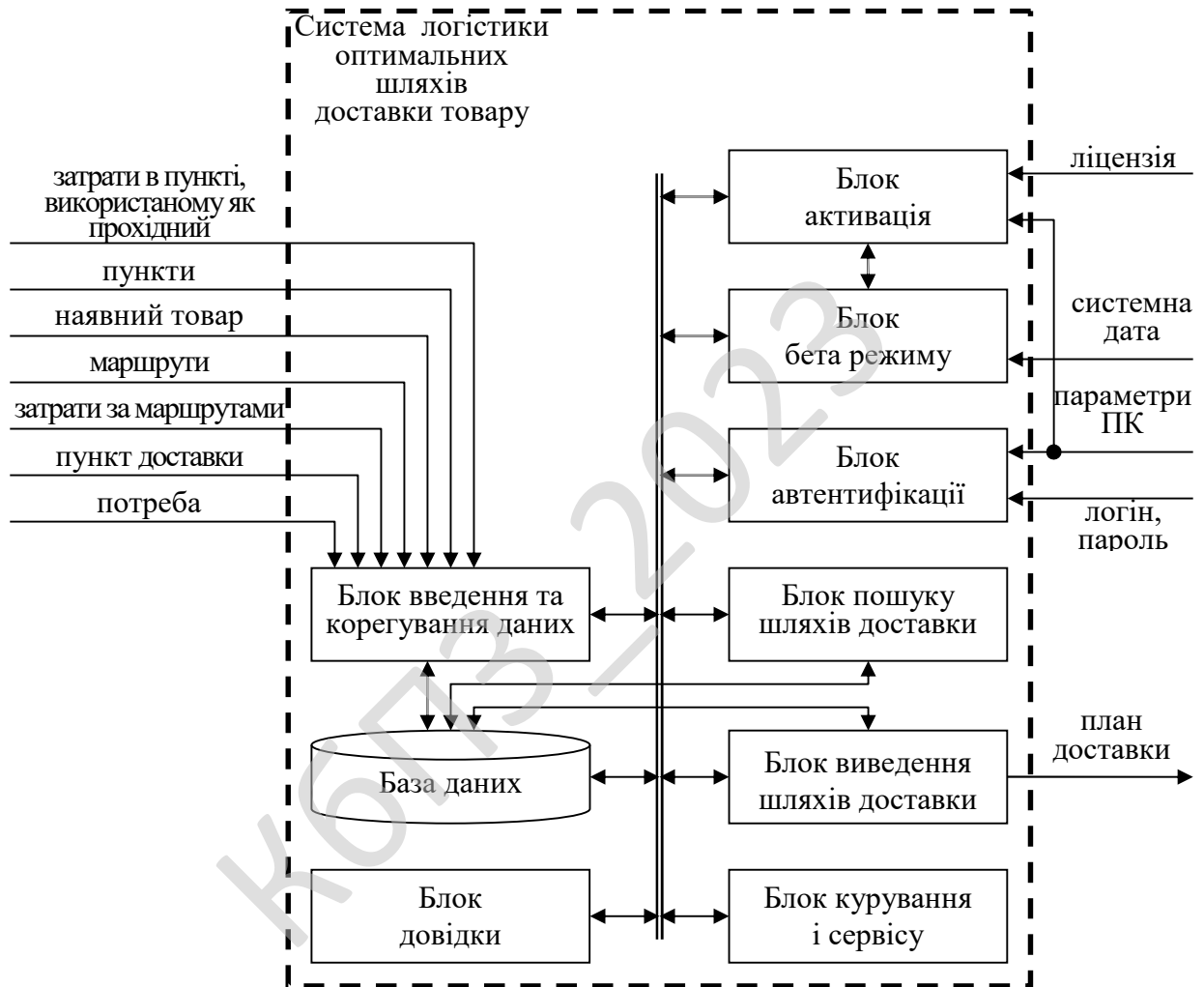


Рисунок 3.1 - Структурна схема

- ліцензія - дані для пропису активації програми;
- системна дата - дані для керування бета (демо) режимом;

- зв'язок між блоками активації та довідки;
- зв'язок між блоками бета режиму та довідки;
- зв'язок між блоками автентифікації та довідки.

3.3 Розробка функціональної схеми

Функціональна схема розробленої системи зображена на рисунку 3.2.

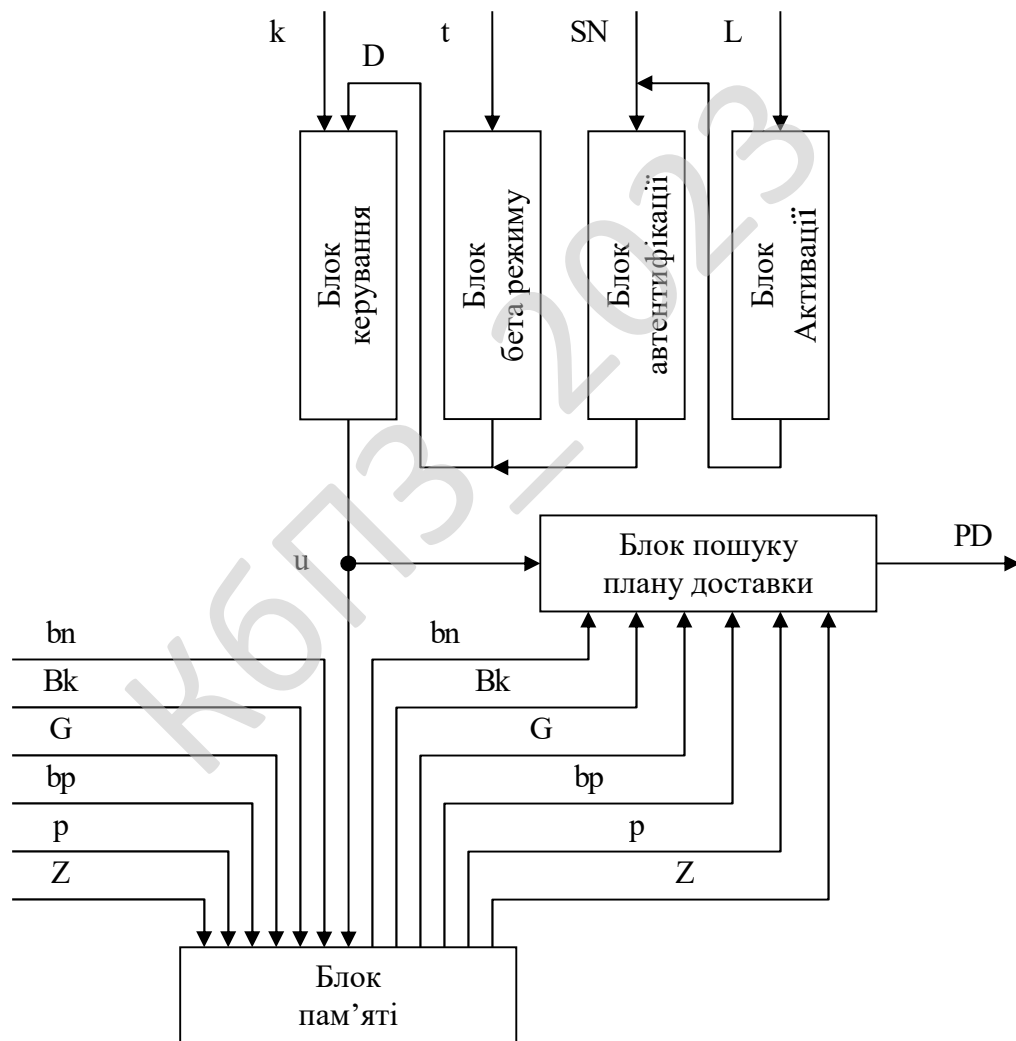


Рисунок 3.2 - Функціональна схема

З рисунка видно, що роботу розробленої системи можна представити функціонуванням сукупності частин: блоку керування; блоку пошуку плану доставки, блоку пам'яті; блоку активації, блоку автентифікації, блоку бета режим, бази даних.

Розглянемо їх роботу.

На початку роботи до блоків активації, автентифікації та блоку бета режим надходять: t - системної дати (по запиту з таймера), SN - серійного номера обладнання ПК (по запиту до обладнання, на який зав'язана ідентифікація ПК) та L - ліцензії (шифрована послідовність, що визначає правомірність працездатності системи на визначеному ПК), за яким формується D (вектор з двох складових A - Автентифікація та B - бета режим, які є логічні величинами), дозвіл на відпрацювання системи.

Після отримання дозволу D блок керування по керуючим впливам k формує керування u (керування на відпрацювання режимів функціонування - номер режиму).

Для режимів введення кодування та корегування даних до пам'яті (бази даних) заносяться дані: bn - кількість вершин; V_k - наявність по пунктах об'єму товару, що доставляється (являє собою одновимірний числовий масив розмірністю bn); направлений граф G , яким описано шляхи та затрати доставки товару з пункт ta в пункт tb (являє собою двовимірний масив розмірністю $bn \times bn$), br - пункт, в якому наявна потреба в товарі (номер кінцевого пункту доставки); p - об'єм потреби товару в кінцевому пункті доставки (числова величина), Z - затрати на використання проміжного пункту (являє собою одновимірний числовий масив розмірністю bn).

Введені дані до пам'яті в режимі пошук оптимального шляху переміщуються до блоку пошуку плану доставки, де перетворюються в PD - план доставки, (являє загалом собою двовимірний масив розмірністю $bn \times 2$, де

в першому стовпці - об'єм доставки, в другому - шлях доставки - перелік пунктів слідування товару та їх номерів).

3.4 Розробка діаграми процесів

Оскільки для опису архітектури інформаційної системи можна скористатися одним із п'яти видів представлень, кожна з яких є однією з можливих проекцій організації і структури системи і відповідає окремому аспекту її функціонування (вид з погляду прецедентів використання, вид з погляду проектування, вид з погляду процесів, вид з погляду реалізації, вид з погляду розгортання системи,) то для опису механізмів синхронізації взаємодій, станів і дій скористаємося діаграмою процесів [26].

Діаграма процесів розробленої системи зображена на рисунку 3.3.

Діаграма процесів представлена у вигляді напрямленого графа, що описує переходи від одного процесу до іншого.

Проаналізувавши діаграму процесів, можна визначити безпосередньо послідовність дій системи, що розробляється. Так, процес роботи програми починається з визначення автентифікації, після чого система переходить до вибору режиму та у випадку наявності дозволів на їх виконання можливий їх вибір. Дозвіл на виконання режиму надала автентифікація. Якщо автентифікація не пройшла, то наявна можливість надання Бета (демо) режиму на виконання; якщо і він не проходить, то можливий лише сервісний режим, де запускається активізація продукту, а тоді далі на визначення автентифікації. З сервісного режиму можливий перехід до автентифікації або проходить повернення до вибору режиму.

У випадку, коли вибрано режими введення даних, корекція початкових даних, пошук плану доставки, вивід результатів, можливе лише повернення до вибору режиму, після чого можливе повторне відпрацювання режимів з

									Лист
									36
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

корекцією даних або без такої, лише повторний пошук (перерахунок) оптимального плану доставки .

Завершення роботи проводиться відповідним вибором з вибору режиму.

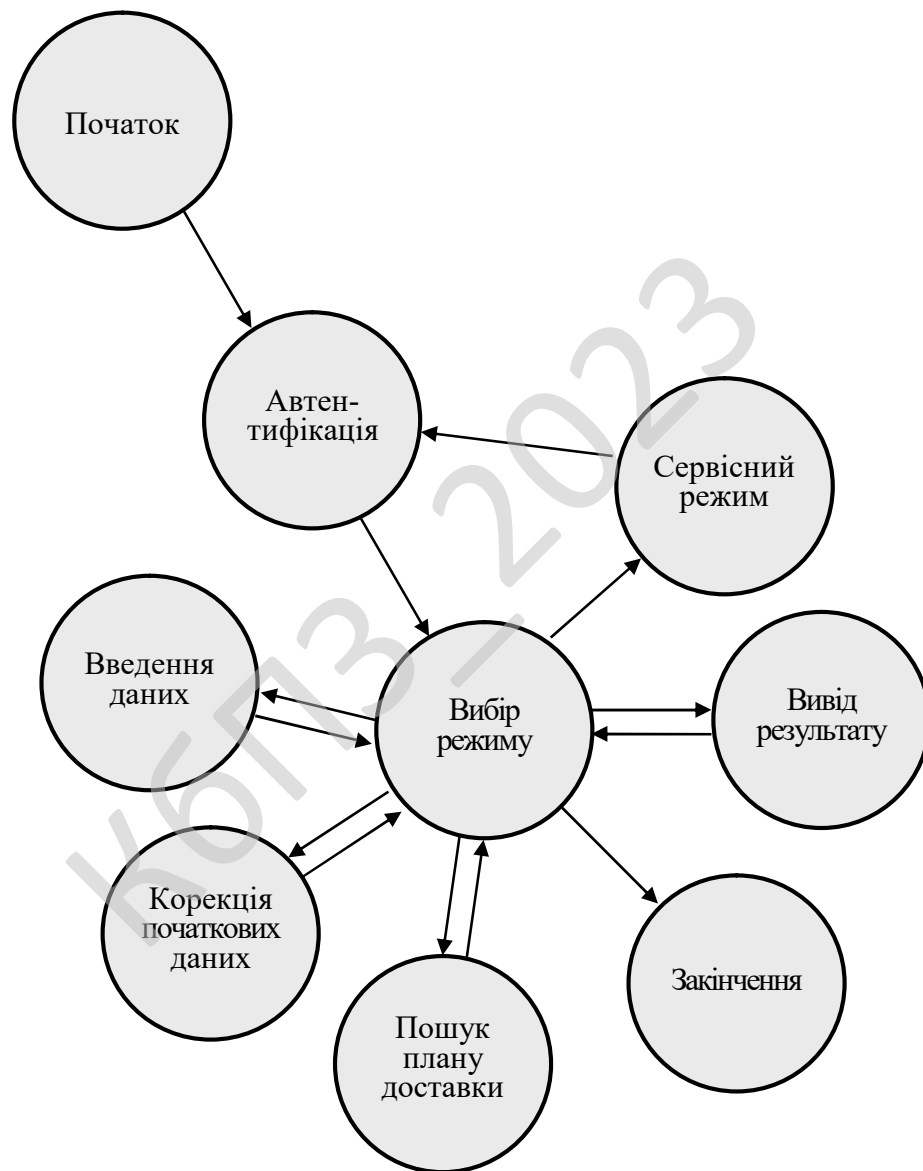


Рисунок 3.3 - Діаграма процесів

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи та діаграму взаємодії процесів, перейдемо до опису блок-схем основної програми та підпрограм, які використовуються для реалізації системи.

КБПЗ – 2023

									Лист
									38
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок–схеми та опис алгоритмів функціонування систем

Блок-схему основної програми наведено на рисунках 4.1 та 4.2 Розглянемо алгоритм роботи основної програми.

З представленої блок-схеми роботи програми видно, що спочатку відбувається перевірка на автентифікацію програми (захист на копію програмного забезпечення), яка проводиться окремою підпрограмою.

Потім здійснюється вибір режиму роботи програми й, в залежності від вибраного режиму, проходить розгалуження виконання роботи програми.

У випадку, коли автентифікацію не проходить - наявна обмежена функціональність (деморезим) доступні тільки сервісний режим для активації та вихід.

При виборі режиму введення даних проводиться введення наступних даних:

b_n - кількість пунктів (окреме числове значення);

V_k - наявність об'єму товару в пунктах постачання (одновимірний числовий масив розмірністю b_n);

G - шляхи та затрати доставки товару з пункту ta в пункт (двовимірний числовий масив розмірністю $b_n \times b_n$),

br – пункт, в якому наявна потреба в товарі (окреме числове значення)

r - об'єм потреби товару в кінцевому пункту доставки (числове значення).

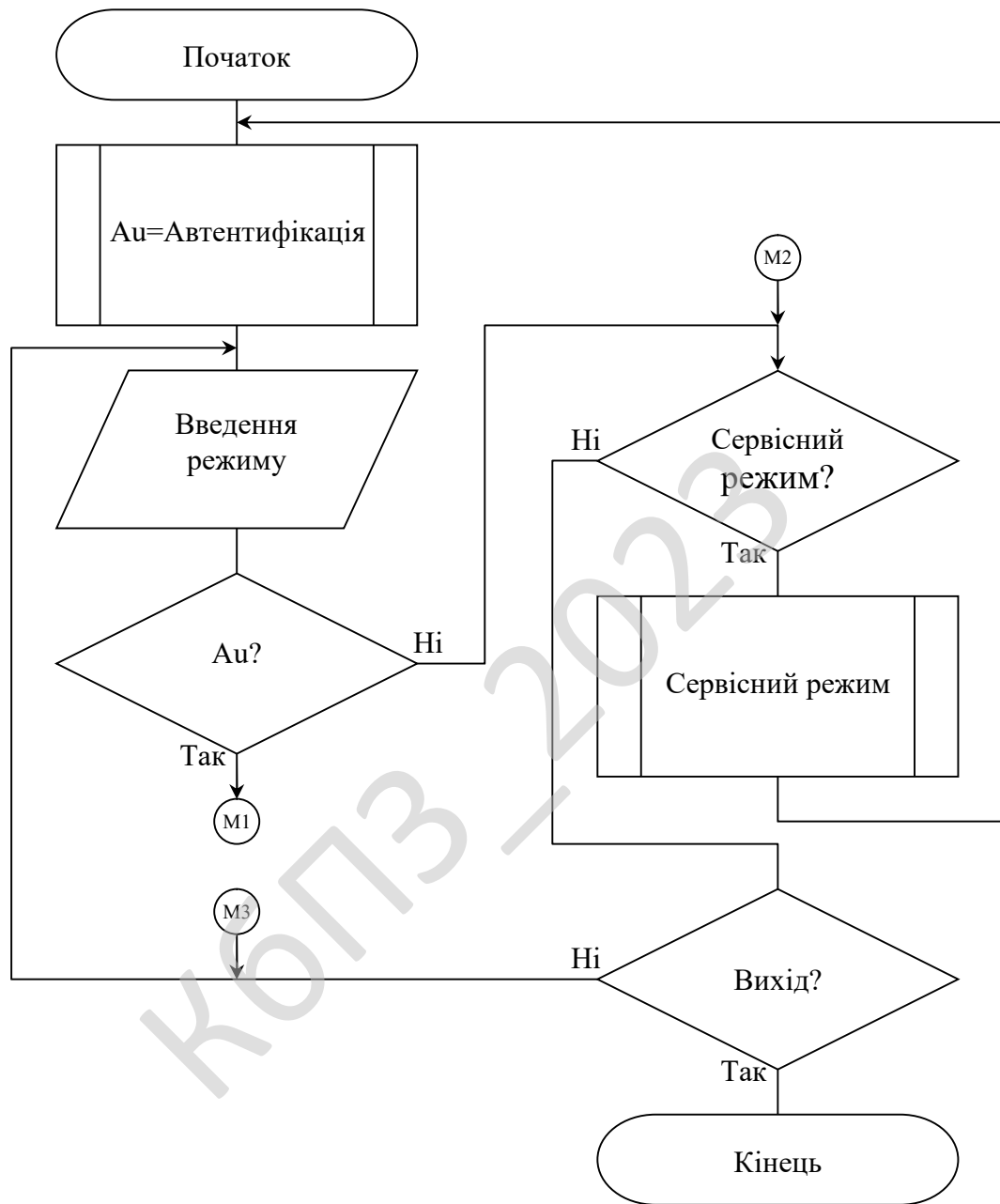


Рисунок 4.1 - - Блок-схема роботи основної програми, частина з перевіркою на автентифікацію, задання режимів, сервісним режимом та організацією виходу.

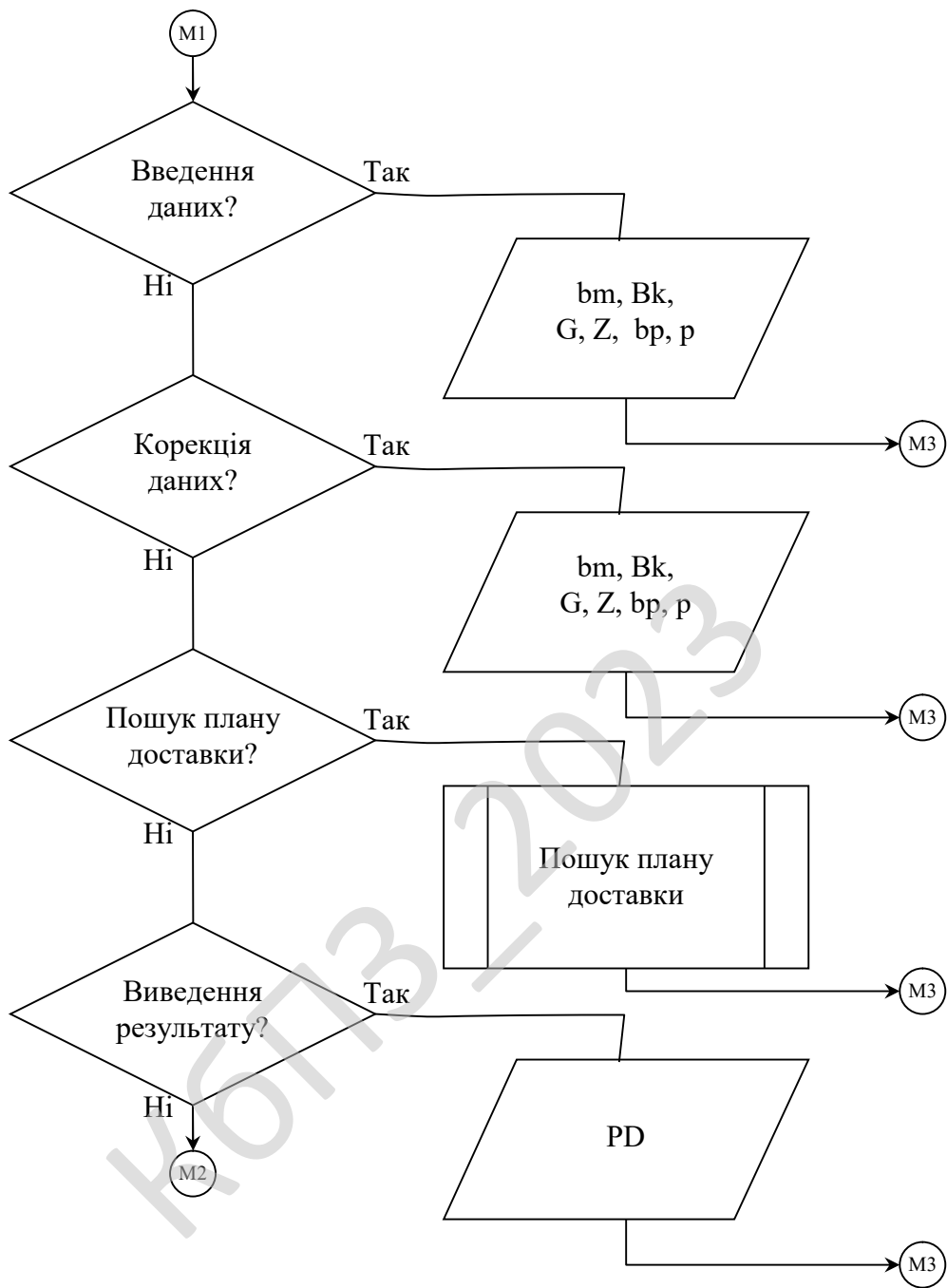


Рисунок 4.2 - Блок-схема роботи основної програми, частина з введенням даних, корекцією початкових даних, викликом пошуку плану доставки та виведенням результату

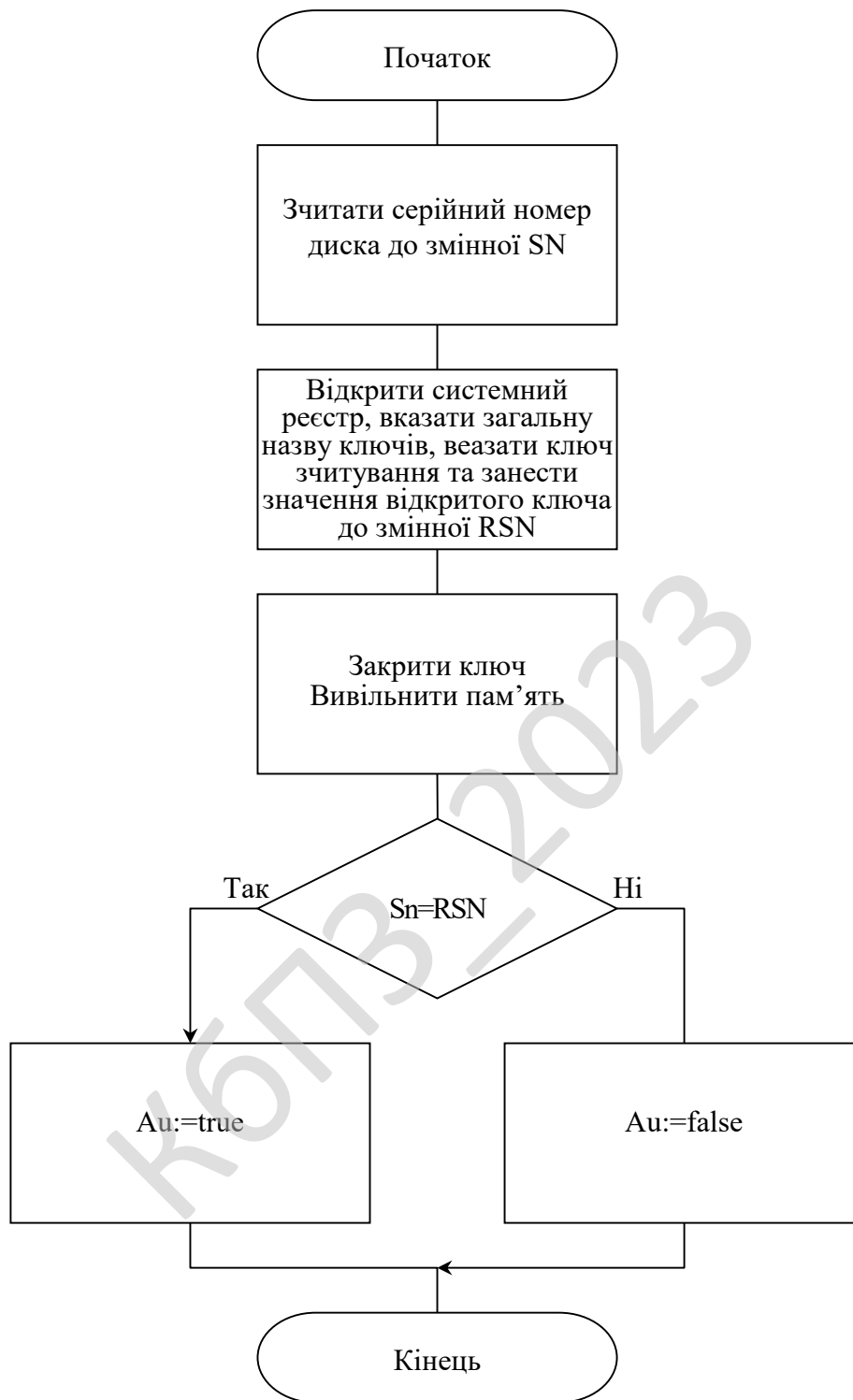


Рисунок 4.3 - Блок-схема роботи підпрограми автентифікації.

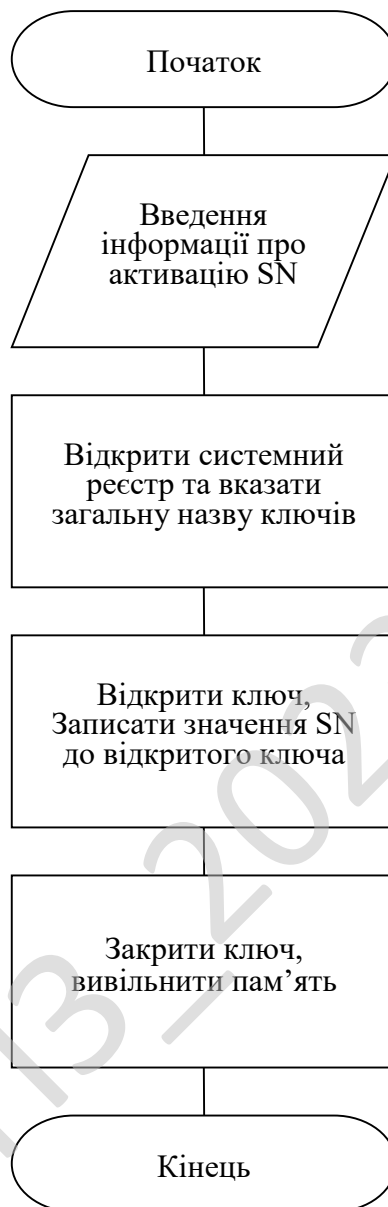


Рисунок 4.4 - Блок-схема роботи підпрограми сервісного режиму (режиму активації).

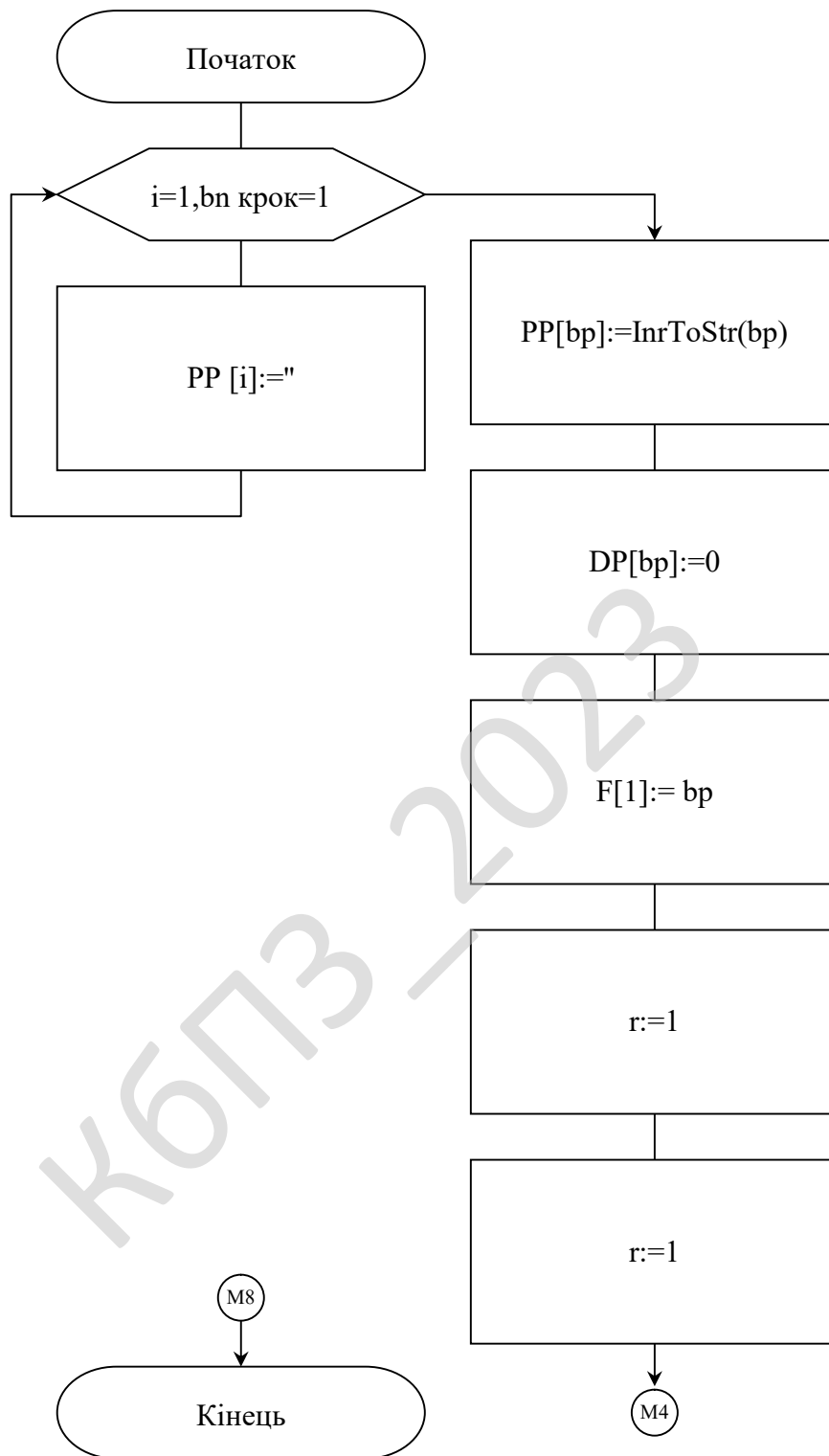


Рисунок 4.5 - Блок-схема роботи підпрограми пошуку оптимального плану доставки, частина з початковими налаштуваннями пошуку

Змн.	Лист	№ докум.	Підпис	Дата

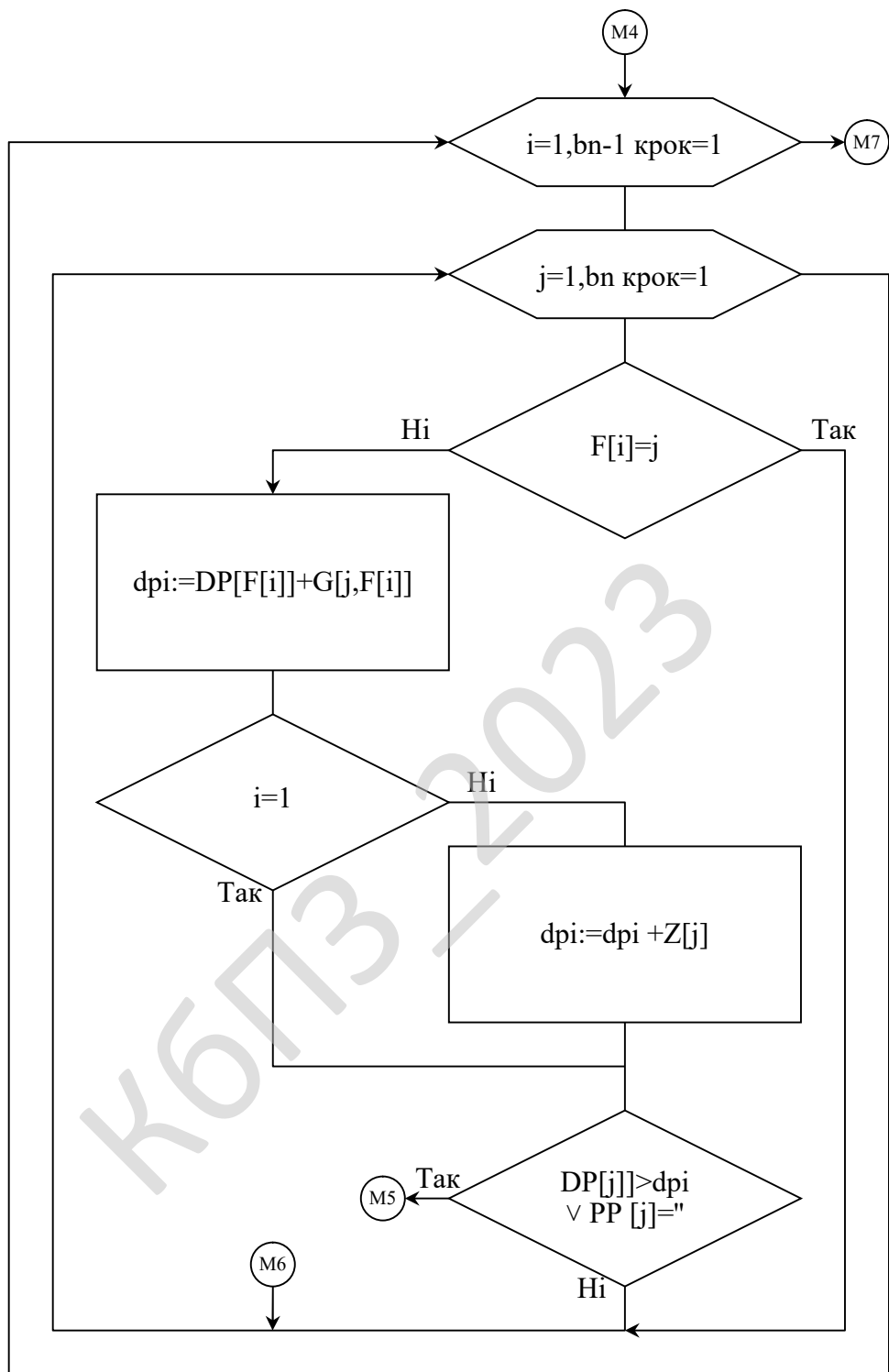


Рисунок 4.6 - Блок-схема роботи підпрограми пошуку оптимального плану доставки, частина з перевіркою можливих шляхів доставки на пошук мінімізації витрат на доставку

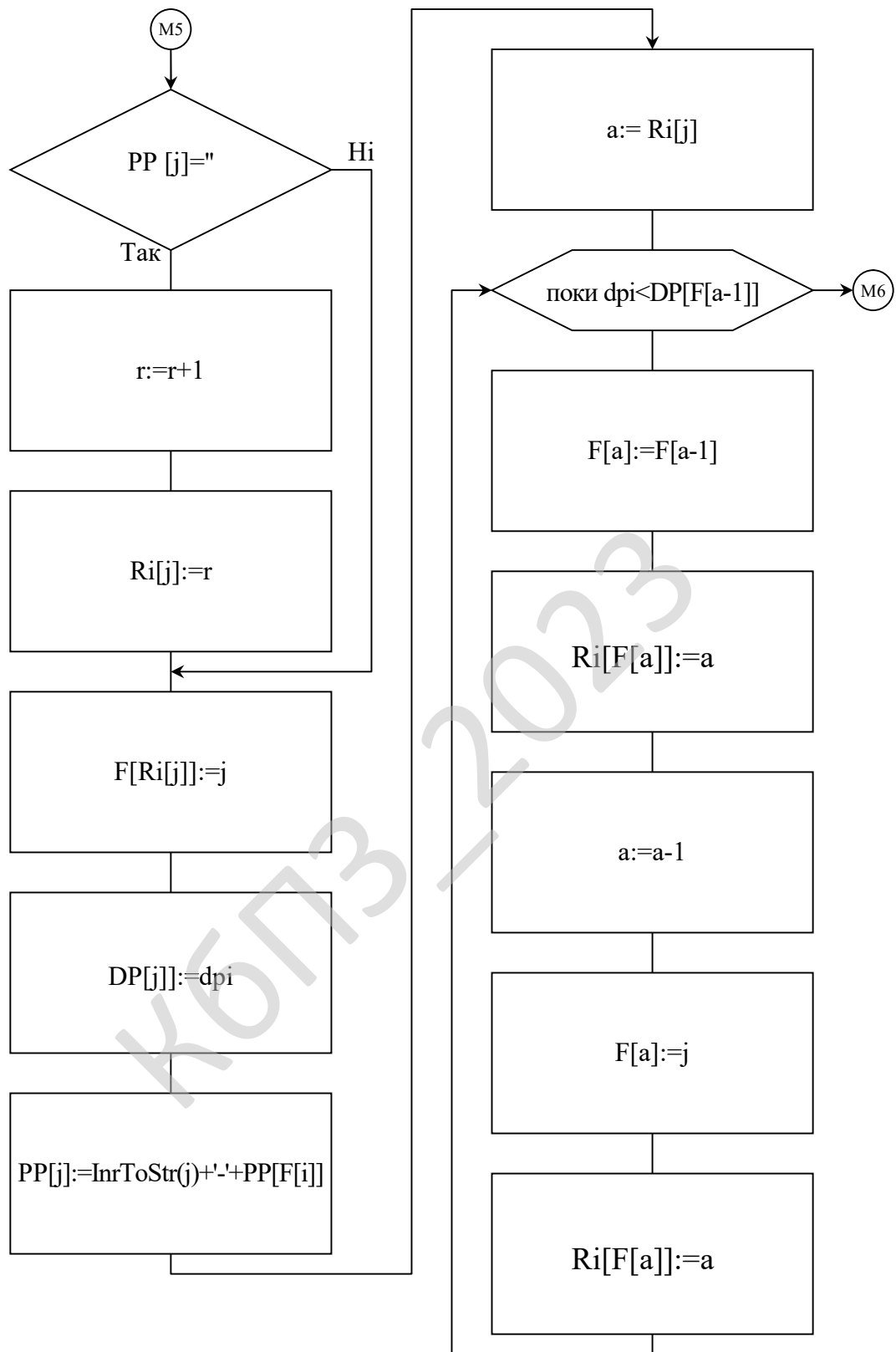


Рисунок 4.7 - Блок-схема роботи підпрограми пошуку оптимального плану доставки, частина з фіксацією віднайденого шляху доставки з пункту

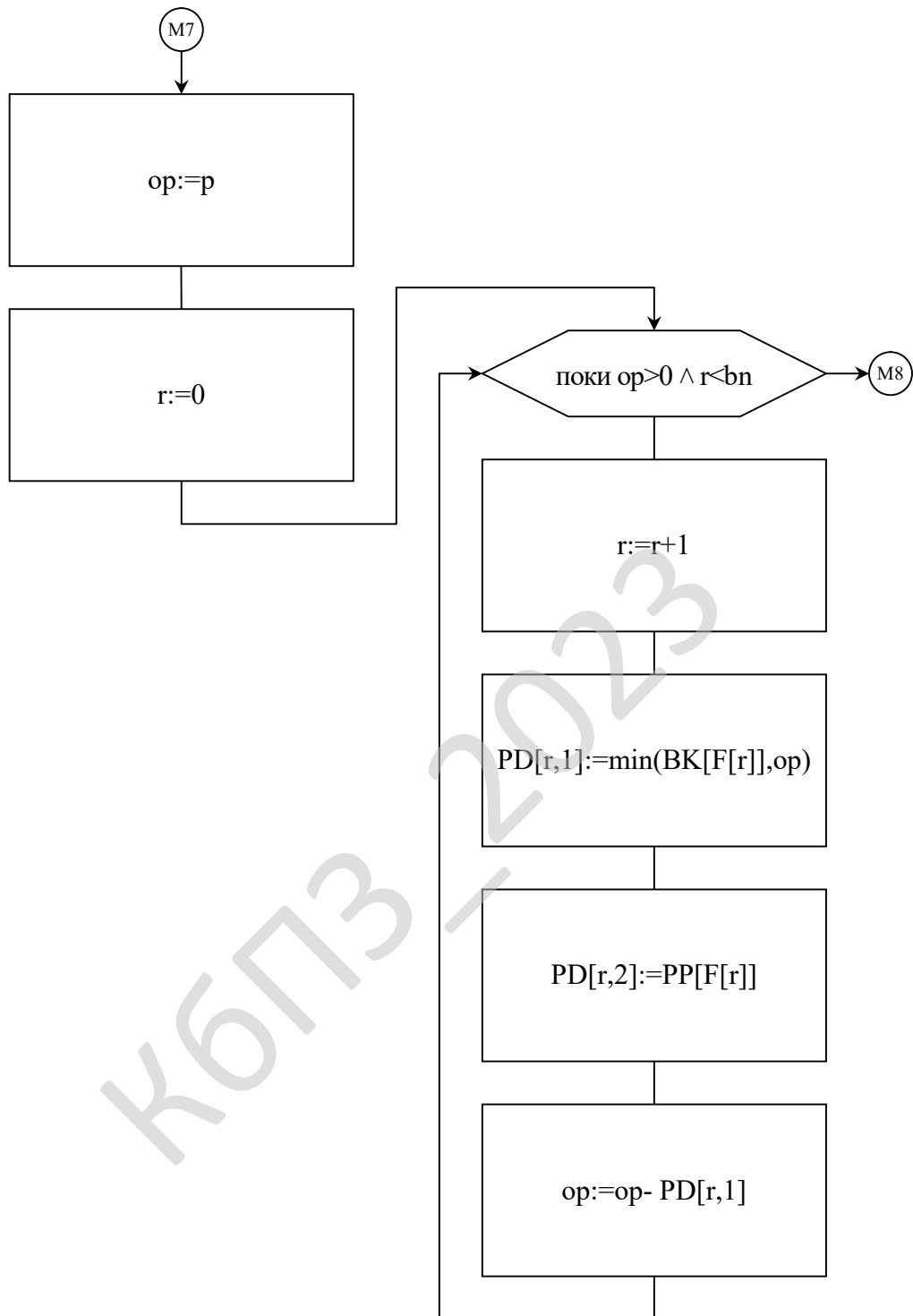


Рисунок 4.8 - Блок-схема роботи підпрограми пошуку оптимального плану доставки, частина з формуванням плану оптимальної доставки товару відповідно до наявних об'ємів та потреб

Наступними діями визначаються для входження в повторну послідовність дій по пошуку доставки до поточного пункту з найближчого в сенсі складності доставки (прямої поставки без використання проміжного пункту) - найменш затратного шляху, наступне кінцевий пункт постачання (визначається номером з переліку повного набору всіх наявних пунктів та записується першим в масив, де буде вміщено в порядку затрат постачання перелік номерів всіх пунктів, які можуть бути задіяні в постачання, як пункти постачальників так і проміжних перевалочних пунктів) та хвіст шляху до нього (хвіст шляху - сам кінцевий пункт). Також попередньо до перебору буде визначено поточні витрати як нульові та початок обходження, як 1 знайдений оптимальний в якості мінімізації затрат пункт.

Наступною ланкою в алгоритмі є організація перебору для знаходження наступного оптимального за затратами найближчого в сенсі складності доставки пункту (яких загалом може бути не більше $b_n - 1$).

Для поточного (кінцевого або проміжного) пункту доставки організується пошук наявних можливих пунктів доставки шляхом організації перебору всіх можливих шляхів до вибраного пункту (загалом їх може бути не більше $b_n - 1$, проте як перебір організовано перебір до b_n лише з виключенням доставки з поточного пункту, як пошуку що не має сенсу). Далі визначаються поточні затрати транспортування. Визначається використання прохідного пункту і для такого, яких знайдено, збільшуються поточні витрати на використання прохідного пункту. У випадку, коли відшукання нового шляху або більш оптимального до пункту не виявлено, проходить повернення до вибору наступного поточного пункту доставки. У випадку, коли такий шлях виявлено, визначається чи є цей шлях до нового, ще незадіяного пункту в схемах доставки, то він додається, збільшується на одну їх кількість, після чого перевизначаються шлях з пункту (виявленого під час цього циклу або раніше) та величина витрат на постачання.

									Лист
									50
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

Наступною сукупність дій в цій ланці упорядкування за затратами ряду пунктів від наступного за поточним до останнього, який використовується (при чому з урахуванням того, що до впорядкованого ряду додається/змінюється всього один елемент, то він просто вставляється по місцю розміщення при зміщенні інших). По завершенню впорядкування пунктів по затратам на доставку також проходить до вибору наступного поточного пункту доставки.

Завершення ланки перебору поточних пунктів настає по закінченні самих пунктів.

Наступною сукупністю дій алгоритму є саме формування плану оптимальної доставки товару відповідно до наявних об'ємів та потреб. Спочатку визначається залишкова потреба як вся потреба. Потім по одному з впорядкованого ряду за затратами, починаючи з першого, додаються до плану шляхи доставки, при цьому зменшується залишкова потреба на наявну можливість поставки шляхом, що додається аж до повного обнулення залишкової потреби або до закінчення можливих шляхів поставок (кожне врахування шляху супроводжується підрахунком кількості використаних шляхів - збільшенням їх лічильника на один).

4.2 Захист розробленого програмного забезпечення

Комерційна реалізація програмного забезпечення потребує захисту останнього (в іншому випадку губиться зміст розробки, як комерційної діяльності працюючого, що направлена на отримання винагород). Питання, щодо захисту можна розглядати з позицій правового регулювання (захисту) та технічного розв'язку. Зупинившись на технічній складовій, відповідно до розповсюдження програмних продуктів, в основу захисту загалом покладено використання програмних та апаратних засобів.

									Лист
									51
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

В якості реалізації пропонується користатися серійним номером жорсткого диску комп'ютера, на якому встановлена програма.

Для визначення серійного номера жорсткого диску засобами Delphi можемо скористатися підпрограмою функцією [28]. Приводимо програмний код, за допомогою якого її реалізовано:

```
function TProtect.GetHDDSerial:dword;  
var  
  a,b,SerialNum:dword;  
  VolumeName : array [0..255] of char;  
begin  
  Result := 0;  
  if GetVolumeInformation(PChar('c:\'), VolumeName, SizeOf(VolumeName),  
  @SerialNum, a, b, nil, 0)  
  then Result := SerialNum;  
end;
```

Оскільки, у відповідності до прийнятої стратегії захисту, зчитаний серійний номер жорсткого диску комп'ютера має порівнюватись з деяким наперед встановленим значенням (у нашому випадку - таємно збереженим в системному реєстрі), то наявна необхідність в його зчитуванні, яка може бути реалізована засобами Delphi, в якості підпрограми функції [28,29].

Приводимо програмний код, за допомогою якого її реалізовано:

```
function RegSerialNamder:dword;  
uses Registry;  
var reg:TRegistry;  
begin  
  reg:=TRegistry.Create;  
  reg.RootKey:=HKEY_LOCAL_MACHINE;  
  reg.OpenKey('HardWareSN', false);  
  Result:=RegKeyGetDw(HKEY_LOCAL_MACHINE;'HardWareSN');  
  reg.CloseKey;  
  reg.Free;  
end;
```

Для реалізації вибраної стратегії захисту наявна необхідність в прихованому доступі на запис до системного реєстру, який також можна провести засобами Delphi, в якості підпрограми функції [28]. Проектну

реалізацію прихованого запису до системного реєстру подамо у вигляді програмного коду:

```
procedure Sn_HDD;
uses Registry;
var reg:TRegistry;
    SerialNum,a,b:dword;
    VolumeName:array [0..255] of char;
begin
    if GetVolumeInformation(PChar('c:\'), VolumeName, SizeOf(VolumeName),
@SerialNum, a, b, nil, 0)
    then
    begin
        reg:=TRegistry.Create;
        reg.RootKey:=HKEY_LOCAL_MACHINE;
        reg.OpenKey('HardWareSN', false);
        RegKeySettDw(HKEY_LOCAL_MACHINE,'HardWareSN',SerialNum);
        reg.CloseKey;
        reg.Free;
    end;
end;
```

Спираючись на значні надбання в сфері захисту програмного забезпечення, зауважимо, що на даному етапі розвитку комерційне використання програмного забезпечення потребує постійного удосконалення технологій та засобів захисту в цілому [30.31, 32,33, 34, 35, 36, 37, 38. 39]. Аналізуючи конкретну реалізацію захисту та наявні тенденції, зазначимо, що в перспективі, при наявності матеріальних можливостей, реалізоване проектне рішення достатньо гнучке для переведення на стратегії захисту, які базуються на використанні електронних ключів.

									Лист
									54
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Загалом використання програмного забезпечення вимагає розгляду наступних завдань: підготовка користувачів програмного забезпечення, встановлення програмного забезпечення, його запуск та налагодження, контроль працездатності та супровід програмного забезпечення.

Підготовка користувачів програмного забезпечення для системи логістики оптимальних шляхів доставки товару загалом не вимагає ніяких специфічних навиків. Програма оснащена довідковою системою її користувача, що має загальні навички роботи з комп'ютерними програмами, без будь-яких труднощів він може одержувати необхідну для роботи інформацію, щодо функціональних можливостей та порядку їх використання. Тож перед введенням в експлуатацію системи логістики оптимальних шляхів доставки товару немає необхідності у проведенні спеціальних навчань персоналу, а тому вони і не проводяться .

Порядок встановлення системи логістики оптимальних шляхів доставки товару не відрізняється від загально прийнятих для програмного забезпечення - для запуску програми необхідно й достатньо організувати доступ до файлів програми, які можуть бути записані на жорсткий диск комп'ютера (перенесені, як звичайні файли, які були скопійовані з іншого комп'ютера із встановленою програмою; інстальовані з дистрибутива - спеціально скомпонованого файлу для завантаження; розпаковано з архіву, скопійовано з мережі тощо) або принесені (монтовані) на зовнішньому з'ємному носію.

Запуск та налагодження системи логістики оптимальних шляхів доставки товару є стандартним для програм, написаних для сімейства ОС

					Лист
					55
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ

Windows. Враховуючи те, що файлом програми є SLODT.exe, запуск програми може проводитися так:

- з меню «програми» операційної ОС (якщо прописано); з меню «виконати» операційної ОС;
- з робочого столу ярликом програми (якщо встановлено сам ярлик на робочому столі для програми);
- з панелі задач (якщо встановлено ярлик на панелі швидкого запуску);
- шляхом натиснення комбінацій гарячих кнопок для запуску програми (якщо така комбінація прописана для даної програми);
- автоматично при завантаженні ОС (якщо вказано автоматичне завантаження при запуску ОС);
- автоматично при підключенні зовнішнього з'ємного носія (якщо автоматичний запуск програми прописано);
- з файлових менеджерів при запуску файла програми тощо.

Після запуску програми отримання доступу до повного функціоналу програми необхідно провести активацію програми, ввівши код активації, в протилежному випадку функціонування програми обмежене, рисунок 5.1.

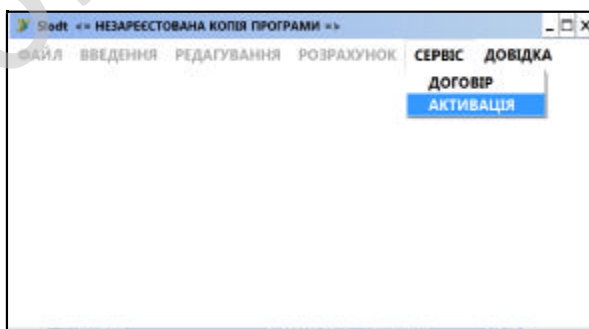


Рисунок 5.1 - Інтерфейс програми, головне меню з обмеженим доступом

Код активації співпадає з номером договору на використання програмного продукту, до якого наявний доступ через меню "СЕРВІС/ДОГОВІР", рисунок 5.2.

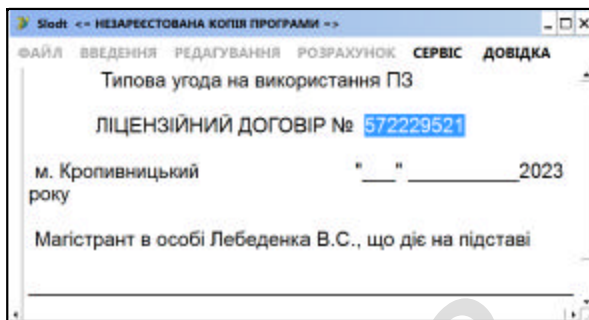


Рисунок 5.2 - Інтерфейс програми, вікно ліцензійного договору

Введення коду активації доцільно провести шляхом копіювання рисунок 5.3,

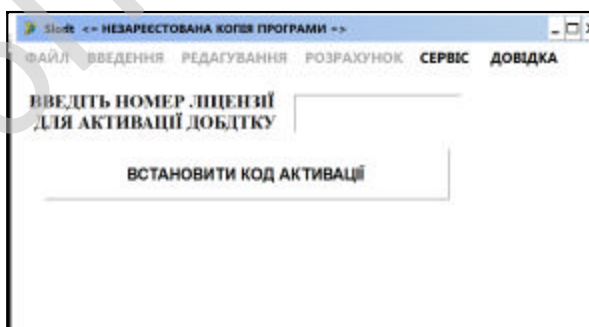


Рисунок 5.3 - Інтерфейс програми, вікно активації програми

Введення коду активації приводить до відкриття доступу до повного набору функцій програми, рисунок 5.4.

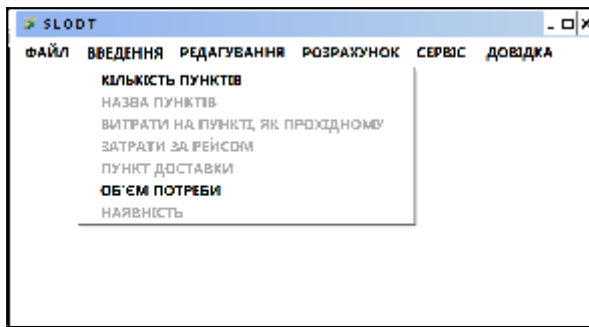


Рисунок 5.5 - Інтерфейс програми, група меню введення з протокольним обмеженням на впорядкованість подання даних

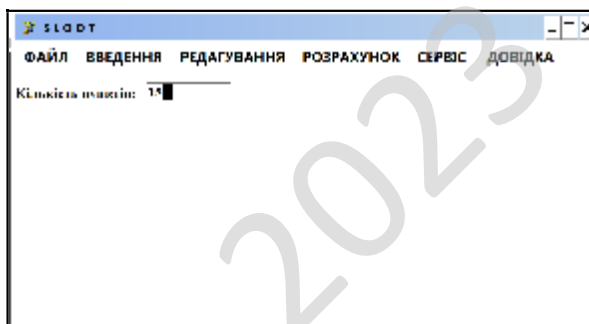


Рисунок 5.6 - Інтерфейс програми, вікно введення кількості пунктів

Після вводимо самі пункти, рисунок 5.7.

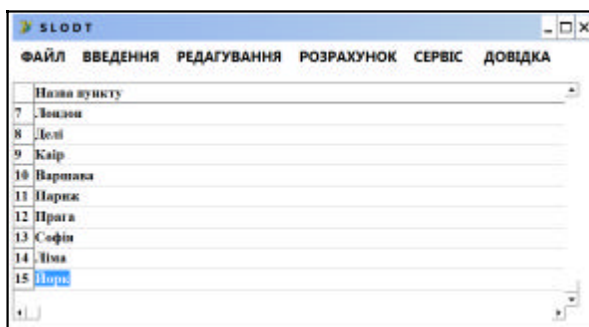


Рисунок 5.7 - Інтерфейс програми, вікно введення пунктів (їх назв)

Наступним кроком вводимо значення витрат, які отримують при використанні пункту, як прохідного, рисунок 5,8.

№	Назва пункту	Витрати на пункт, як прохідному
1	Київ	12
2	Одеса	9
3	Полтава	8
4	Рівні	11
5	Берлін	21
6	Осао	51
7	Лондон	12
8	Делі	41
9	Каїр	23
10	Варшав	12
11	Париж	12
12	Прага	12

Рисунок 5.8 - Інтерфейс програми, вікно введення витрат на пунктах , при використанні їх як прохідних

Як складові, вводяться прості маршрути у вигляді двовимірного поєднання, що нагадують такі ж масиви (або матриці), рисунок 5.9.

Відправка	Пробуття	1	2	3	4	5	6
4	Рівні	23	10	12		12	
5	Берлін	24	15		12		22
6	Осао	61	50	15		21	
7	Лондон	57	41	45	12	32	20
8	Делі	108	12		220		300
9	Каїр	85	150	102	102	11	122
10	Варшав	15	23	12	12		12
11	Париж	14	52	154	10	21	21
12	Прага	10	12	12	20		

Рисунок 5.9 - Інтерфейс програми, вікно введення маршруту

З наявних (раніше введених пунктів) обираємо пункт доставки - просто з наявного переліку, вибравши потрібний, рисунок 5.10.

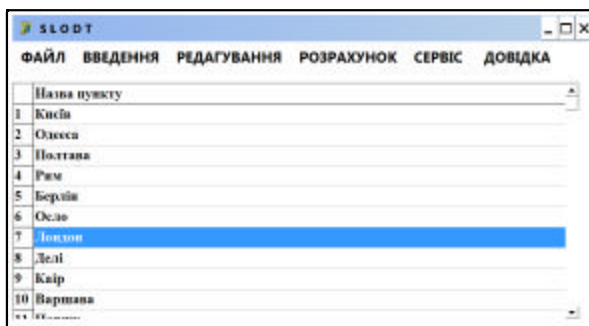


Рисунок 5.10 - Інтерфейс програми, вікно вибору пункту призначення

Далі визначаємо наявність об'єму товару в пунктах, з яких передбачається його доставка, рисунок 5.11.

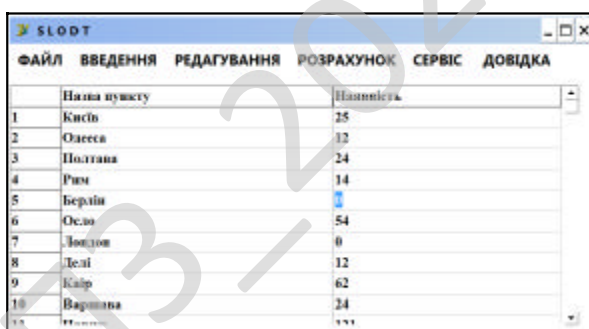


Рисунок 5.11 - Інтерфейс програми, вікно введення розподілу товару по пунктах

Й наостанок для введення визначається обсяг потреби в пункті призначення, рисунок 5.12.

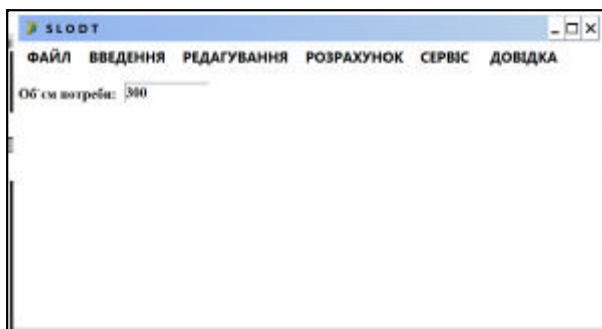


Рисунок 5.12 - Інтерфейс програми, вікно введення об'єму потреби товару в кінцевому пункті

Оскільки ступінь обчислювальної складності задачі в залежності лише від вхідних даних великий - одних тільки рейсів може бути більше 200 лише при 15 пунктах, які задіюються при доставці, то наявна можливість збереження даних суттєво спрощує використання програми, рисунок 5.13

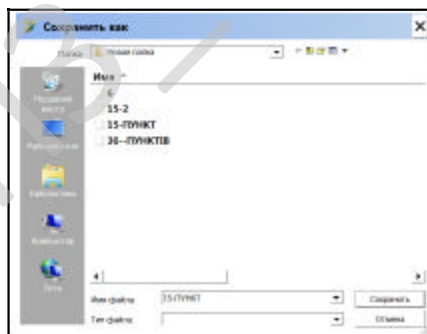


Рисунок 5.13 - Інтерфейс програми, вікно збереження введених даних

Також відповідно наявна можливість і відновлювати дані, рисунок 5.14

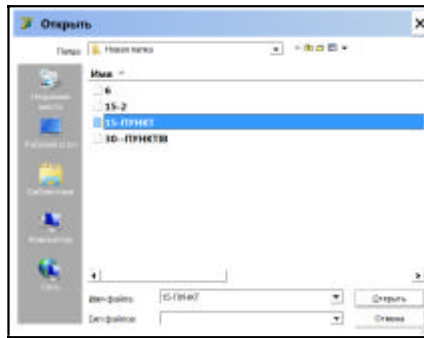


Рисунок 5.14 - Интерфейс програми, вікно завантаження даних з файлу

Після завантаження даних будь-яким доступним способом наявні можливості їх аналізу:

- з позиції побудови найкоротших шляхів доставки з усіх наявних пунктів з розташуванням у пріоритеті ціни доставки, рисунок 5.15:
- з позиції доставки під потребу без розгляду затратніших шляхів (затратні з позиції пункту, який потенційно несприятливий), рисунок 5.16;
- з позиції розпису складових витрат для кожного з наявних оптимальних шляхів доставки з обраного пункту, рисунок 5,17.

Пункт	Об'єм	Ціна	Шлях
1	0	0	Лондон
2	25	12	Київ->Лондон
3	54	12	Одеса->Лондон
4	123	12	Париж->Лондон
5	24	22	Варшава->Лондон
6	12	22	Прага->Лондон
7	12	25	Одеса->Київ->Лондон
8	24	27	Полтава->Київ->Лондон
9	2	33	Софія->Київ->Лондон
10	14	39	Будапешт->Париж->Лондон
11	14	44	Рим->Одеса->Київ->Лондон
12	12	46	Делі->Одеса->Київ->Лондон
13	12	47	Лима->Варшава->Лондон
14	0	48	Берлін->Київ->Лондон
15	62	55	Каїр->Прага->Лондон

Рисунок 5.15 - Интерфейс програми, вікно розрахунку для знаходження повного набору всіх можливих мінімальних за затратами шляхів постачання



Рисунок 5.16 – Інтерфейс програми, вікно розрахунку для знаходження набору мінімальних за затратами шляхів постачання заданого об'єму потреби

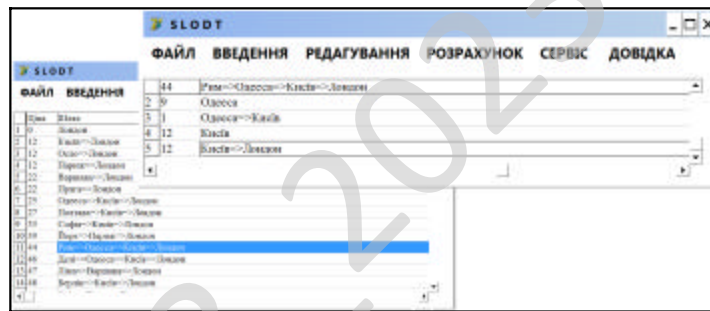


Рисунок 5.17 – Інтерфейс програми, вікно аналізу витрат затрат по вибраному шляху постачання товару

Зауважимо, що, незважаючи на можливі некоректні зміни умов пошуку, наявні й редагування вхідних даних, проте вони винесені окремо і можуть працювати як введення, але незначні зміни можуть привести до втрати великої кількості даних. Тому початкове введення не передбачає просту зміну в режимі початкового введення, а винесено в окремий режим корегування, який доцільно проводити лише після збереження даних, оскільки існує велика ймовірність втрати введених даних. По суті і

зовнішньому вигляду вони нічим не відрізняються візуально, наявне лише блокування на знищення введених попередніх даних.

Також наявні блокування при введенні і корегуванні, які покликані захистити на введення неіснуючих значень, що можуть привести до збою помилок розрахункових методів (наприклад: введення символічних значень замість числових, введення некоректних за типом числових значень і тому подібне).

Наостанку зазначимо наявність довідкового вікна з короткою інформацією про продукт проектування, рисунок 5.18.

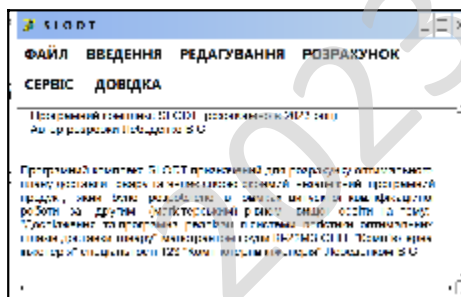


Рисунок 5.18 – Інтерфейс програми, вікно довідки про програму

Успішність проекту на царині ІТ технологій передбачає обов'язковий супровід програмного забезпечення з контролем працездатності, який призначений забезпечувати постійний прибуток шляхом підтримання зростаючих вимог щодо функціональних можливостей програмних засобів. Зважаючи на зазначене, для успішності розробки передбачена можливість програми для модернізації - додавання нових методів побудови оптимальних шляхів. Внаслідок виправлень, виявлених у процесі експлуатації помилок, процес підтримки працездатності системи логістики оптимальних шляхів доставки товару приводить до внесення зміни в програму, що в підсумку виліється в перехід до нових її версій.

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення для сфери логістики доставки товару, яке призначено для зниження витрат на перевезення товару при наявній умові збереженості його якості.

Метою розробки є дослідження та програмна реалізація системи логістики оптимальних шляхів доставки товару.

Об'єктом дослідження є процес пошуку оптимальних шляхів доставки товару.

Предметом дослідження є методи пошуку оптимальних шляхів доставки товару.

Методи дослідження базуються на методах пошуку оптимальних шляхів, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі вирішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод пошуку оптимальних шляхів доставки товару.
- Розроблено вітчизняний продукт пошуку оптимальних шляхів доставки товару, який має більш широкі можливості, на відміну від існуючих аналогів.

									Лист
									66
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

Продовження таблиці 7.1

1	2	3
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість, відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2

Продовження таблиці 7.1

1	2	3
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	28000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

Змн.	Лист	№ докум.	Підпис	Дата

різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-
дні:

$$T_{PI} = 0,3CT_{\text{уточн}}^{0,33+0,2(B-1,01)}S, \quad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ, згідно з встановленими вимогами.

Вибираємо в межах (25...350)%.

$$TPI = 0,3 \cdot 2,66 \cdot 9,370,33 + 0,2(1,026 - 1,01) \cdot 38 = 64 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	64	Ф 7.1-7.4
Впровадження	13	Д13
Всього	105	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{пз} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;

$T_{пз}$ – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{105 \cdot 1}{60 - 5} = 1,9 \text{ ставки.}$$

Чисельність інженерів-електронників для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт протягом року.

Визначаємо затрати часу на виконання профілактичних робіт з обслуговування обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронників не повинен складати більше 10%.

Виходячи з цього, фонд робочого часу інженерів-електронників складає:

$$\Phi_{op}^c = \frac{3_ч \cdot n_{mic}}{1,2}, \quad (7.6)$$

$$\Phi_{op}^c = \frac{60,74 \cdot 3}{1,2} = 151,85 \text{ год.}$$

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт з обслуговування обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на од. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	11	990	16,5
Монітор	60	11	660	11
Клавіатура	30	11	330	5,5
Маніпулятор «мишка»	30	11	330	5,5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	1	120	2
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор– маршрутизатор	30	4	120	2
Кабельні господарства ЛВС на 1 м. п.	2,5	350	875	14,58
Копіювальний апарат	140	1	140	2,33
Усього за рік:	3 _ч	60,74		

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$Ч_{ел} = 151,85 / (60 \cdot 8) = 0,3 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронників.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2022, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN, PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Розміщення графіки і контенту на Інтернет сторінках	0,5	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додруковка підготовка макетів	0,25	
	Розміщення графіка і контенту на Інтернет сторінках	0,25	
Всього		2	

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	21760	65280
Продакт-менеджер	0,25	18000	13500
Інженер-програміст	1,9	20000	114000
Інженер-електронник	0,3	15000	13500
Інженер-системотехнік	0,25	15000	11250
Адміністратор мережі	0,5	15000	22500
Системний програміст	0,25	15000	11250
Дизайнер WEB	0,25	17000	12750
Інженер-верстальник	0,25	15000	11250
Бухгалтер-економіст	0,5	20000	30000
Всього за період розробки	$R_{cn} = 5,45$	-	$\Phi_{роб} = 305280$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де: $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{305280}{6,35 \cdot 60} = 800 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

$$B_{y\delta} = R_{cn}^1 S_y \Pi_{пл}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

S_y – питома площа на одне робоче місце, m^2 ;

$\Pi_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно з даними ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 400...1600 $у.о./m^2$. Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно $8m^2$. З урахуванням цього:

$$Буд = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{нв} = R_{cn}^1 \cdot \Pi_m, \quad (7.10)$$

де: Π_m – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 8 \cdot 3500 = 28000 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається за оптовими цінами постачальника з урахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Компбест за 08.11.23 – джерело <https://compbest.com.ua/>.

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптов а ціна
Персональний комп'ютер		10947
Системний блок		7347
Процесор	INTEL Core™ i3 10105 (BX8070110105) 1200, 4 ядра, 8 потоків, 3.7 GHz, 4.4 GHz, TDP - 65 Вт, 14nm, BOX	-
Системна плата	ASUS PRIME H510M-K сокет - 1200, DDR4, 64 ГБ, 3200 MHz, LAN - 1 Гбит/с, D-Sub (VGA), HDMI, 1 x M.2 2280, 4 x Sata 6.0 Gb/s, Micro-ATX	-
Відеокарта	Intel UHD Graphics 630	-
Жорсткий диск	SSD M.2 2280 240GB Apacer (AP240GAS2280P4-1) 240 GB, 3D TLC, M.2, PCI Express 3.0 x4	-
Оперативна пам'ять	DDR4 8GB 3200 MHz Fury Beast Black Kingston Fury (KF432C16BB/8)	-
Блок живлення	Gamemax 500W (GM-500B) ATX 12V v2.3, 500 Вт, 20+4 pin, CPU - 4+4pin, GPU - 1x6 pin, SATA - 3, Peripheral - 2, +12V1 - 20A, 1x120 мм, 150 x 140 x 86 мм	-
Корпус	Vinga CS210B, Miditower, ATX, Micro - ATX, Mini - ITX	-
Кулер	-	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	220
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 1, 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	15	10947	16420,5	180625,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	-	-	-	0
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	199177

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	199177	-	-
Всього по групі	199177	50	99588,5
Група 5, 6			
4. Вимірювальні пристрої	5190	25	1297,5
5. Транспортні засоби	0	20	0,0
6. Господарський інвентар	28000	25	7000
Всього по групі	33190	-	8297,5
7. Нематеріальні активи	120000	10	12000
Разом	$K_p = 1760367$		$A_p = 190286$

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 800 \cdot 105 / 280 = 300 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 300 \cdot 10 \cdot 0,01 = 30 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $H_c = 22\%$ від суми основної та додаткової зарплати:

$$C_{oi} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де: H_c – відрахування на соціальні потреби, %.

$$C_{oi} = 0,01 \cdot 22(300+30) = 73 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_g = 15\%$ від основної зарплати:

$$G_{ocn} = Z_o \cdot H_g \cdot 0,01, \quad (7.14)$$

де: H_g – загальногосподарські витрати, %.

$$G_{ocn} = 300 \cdot 15 \cdot 0,01 = 45 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$ZM = (ZM1 + ZM2 + ZM3) / N_e, \quad (7.15)$$

де: $ZM1$ – вартість паперу, грн.;

ЗМ2 – вартість запам'ятовуючих пристроїв, грн.;

ЗМ3 – вартість фарби, картриджів, тонеру, грн.;

Ne – кількість екземплярів програм, шт.

Згідно виданих норм приймаємо пів пачки паперу на три місяці розробки ($np=0,33$). Тоді, враховуючи, що вартість пачки паперу складає $Цп = 210$ грн., визначаємо вартість паперу за період розробки $Nм = 3$ міс:

$$ЗМ1 = Цп \cdot np \cdot Nм. \quad (7.16)$$

$$ЗМ1 = 210 \cdot 0,5 = 105 \text{ грн.}$$

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків в кількості 100 екземплярів програм:

$$ЗМ2 = \sum Цд, \quad (7.17)$$

де: Цд – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 33,6 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 44 грн./шт.

$$ЗМ2 = 33,6 \cdot 100 = 3360 \text{ грн.}$$

Згідно виданих норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$ЗМ3 = \sum Цз., \quad (7.18)$$

де: Цз – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$ЗМ3 = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$ЗМ = (105 + 3360 + 1702) / 280 = 18 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($Hп = 15\%$) від основної зарплати виконавців:

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: $Hп$ – норматив витрат на освоєння нових мов програмування, %.

									Лист
									80
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

$$O_n = 300 \cdot 15 \cdot 0,01 = 45 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 280$ прим.):

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 190286 \cdot 3 / (280 \cdot 12) = 170 \text{ грн.}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 300 + 30 + 73 + 45 + 18 + 45 + 170 = 681 \text{ грн.}$$

Величина ціни підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.21)$$

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 681 = 341 \text{ грн.}$$

пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.9.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	1226
Всього капітальних витрат	–	1226

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми протягом року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на обслуговування системи	Z_p	12250	4800
2. Витрати на електроенергію	Z_{el}	2692	1864
3. Витрати на амортизацію	$Z_{ам}$	0	307
Всього витрат за рік	I	14942	6971

Витрати на профілактичні роботи:

$$Z_p = T_p \cdot Z_z \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування кожного комп'ютера за рік, год.;

Z_r – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення витрати на обслуговування системи зменшилися з 12250 грн до 4800 грн на рік.

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн., за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	1226	–	306,5
Всього відрахувань	-	–	1226	–	306,5

Витрати на електроенергію визначаються з урахуванням спожитої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$):

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел} \quad (7.24)$$

$$Z_{ел\ баз} = 0,545 \cdot 1300 \cdot 3,8 = 2692 \text{ грн.}$$

$$Z_{ел\ нов} = 0,545 \cdot 900 \cdot 3,8 = 1864 \text{ грн.}$$

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_e = (I_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де: K_p – балансова вартість основних фондів розробника, грн.; E_p – розрахунковий коефіцієнт капіталовкладень.

$$E_e = (1022 - 681) \cdot 280 - (0,05 \cdot 1408000 + 0,5 \cdot 199177 + 0,25 \cdot 33190 + 0,1 \cdot 28000) \cdot 3/12 = 50208 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p}{(I_n - C_n) \cdot N_e}, \quad (7.26)$$

де: K_p – балансова вартість основних фондів розробника.

$$T_e = \frac{1760367}{(1022 - 681) \cdot 280 \cdot 12 / 3} = 4,5 \text{ роки}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\bar{o}} - I_n) - E_n (K_n - K_{\bar{o}}), \quad (7.27)$$

де: $I_{\bar{o}}$, I_n – величина експлуатаційних витрат за базовим і новим варіантом відповідно;

$K_{\bar{o}}$, K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (14942 - 6971) - 0,25 \cdot 1226 = 7675 \text{ грн.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{1226}{14942 - 6971} = 0,15 \text{ року.}$$

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	280
2. Повна собівартість розробленої програми	Грн.	681
3. Ціна розробленої програми	Грн.	1022
4. Плановий прибуток від реалізації розробленої програми	Грн.	341
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1760367
7. Загальний прибуток від реалізації програмної продукції	Грн.	95480
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	50208
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	4,5
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	1226
11. Величина економічного ефекту у користувача програмної продукції	Грн.	7675
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,15

Змн.	Лист	№ докум.	Підпис	Дата

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищується якість приймаючих управлінських рішень.

КБПЗ – 2023

									Лист
									87
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

8 ЗАХОДИ ЩОДО ОХОРОНИ ПРАЦІ І ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Згідно із Законом України про охорону праці, маємо визначення: "Охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі праці", тож в рамках цього визначення маємо, що техніка безпеки - це система правових, соціально-економічних, організаційно-технічних та санітарно-гігієнічних заходів та засобів, спрямованих на збереження життя та здоров'я людини під час її трудової діяльності, тобто це низка заходів по створенню безпечних умов виробництва для працівників під час виконання ними своїх трудових обов'язків, а отже, відповідно державна політика України спрямована на створення належних, безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням. Основою державної політики в цьому питанні є:

- пріоритет життя і здоров'я працівників, повна відповідальність роботодавця за створення належних, безпечних і здорових умов праці;
- підвищення рівня промислової безпеки шляхом забезпечення суцільного технічного контролю за станом виробництва, технологій та продукції, а також сприяння підприємствам у створенні безпечних та нешкідливих умов праці;
- комплексне розв'язання завдань охорони праці на основі загальнодержавної, галузевих, регіональних програм з цього питання та з

										Лист
										88
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ					

"Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями", ДСанПіН 3.3.2.007-98 "Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин" та ДСТУ 8604:2015 "Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги", які призначені для запобігання впливу на працівників шкідливих і небезпечних факторів. До загальноправової бази, що регулюють нормативно-правові положення з техніки безпеки (й для в ІТ-сфері в тому числі), відносяться Інструкції з охорони праці при роботі з електронно-обчислювальними машинами, в яких протокольно описано поведження під час штатних робіт та в аварійних ситуаціях, основним призначенням яких є запобігання наслідків (зменшення або повне їх усунення) небезпек, що пов'язані з виробничою діяльністю.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Якщо послуговуватися класифікацією небезпечних й шкідливих виробничих факторів при роботі на комп'ютері по природі їх виникнення, а отже поділивши їх на наступні групи: фізичні, хімічні, психофізіологічні, біологічні, то до фізичних можемо віднести:

- підвищену й знижену температуру повітря;
- підвищену й знижену вологість повітря;
- недостатню освітленість робочого місця;
- перевищуючі припустимі норми шуму;
- підвищений рівень іонізуючого випромінювання;
- підвищений рівень електромагнітних полів;
- підвищений рівень статичної електрики;
- безпеку враження електричним струмом;

									Лист
									90
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

- бляклість екрана дисплея;
- до хімічних можемо віднести:.
- виникнення, у результаті іонізації повітря при роботі комп'ютера, активних часток;
- до психофізіологічних можемо віднести:.
- розумова напруга;
 - втрата реальності;
 - виникнення залежності;
 - нервово-емоційні перевантаження;
 - перенапруга зорового аналізатора;
- до біологічних можемо віднести:
- бактеріологічну небезпеку, пов'язану з наявністю місць із сприятливим середовищем їх розмноженням (наприклад клавіатура).

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Проаналізуємо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 - Розміри приміщення

Найменування	Значення, м
Ширина	4
Довжина	5
Висота	3

Таблиця 8.2 - Площа та обсяг приміщення, на одного працюючого*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м ²	Не менше 6.0	6,67
Об'єм, V	м ³	Не менше 20.0	20

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

У зазначеному приміщенні працюють троє людей. За даними, які наведено у табл. 8.1, та табл. 8.2, можна зробити висновок, що площа та об'єм приміщення з розрахунку на одне робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 "Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин" [42] але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 "Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями" [42] та НПАОП 0.00-7.15-18 "Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями". Таким чином, можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря

України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Таблиця 8.3 - Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Вологість, %	Швидкість повітря, м/с	Температура, °С	Вологість, %	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	40-55	0,11
Тепла	23-25	50-70	0,1	24-25	50-65	0,17

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер HP LaserJet Pro P1102, електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини та попереджають травматизм і професійні захворювання програмістів, є освітлення на робочому місці.

З 2019 року діють Державні будівельні норми України ДБН В.2.5-28:2018 "Природне і штучне освітлення" [43], у яких прописані вимоги до використання всіх освітлювальних приладів, у т.ч. світлодіодних.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5-28:2018 [43], можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [43], Крім того, все поле зору повинне бути освітлено достатньо рівномірно - це основна гігієнічна вимога. Так, як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

									Лист
									94
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розміри приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень;

Таким чином, можна припустити, що основною причиною зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці й відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

Регулярне наочне знайомство персоналу із шляхами для евакуації людей із приміщення, відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками із заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так, як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

									Лист
									95
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

8.5 Розрахункова частина

Проводимо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення, ширина якого складає – 4 м, довжина – 5 м, висота – 3 м [44,45].

У зазначеному приміщенні працює 3 особи.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою:

$$F = E S K Z / n, \quad (8.1)$$

де: F - світловий потік, що розраховується, Лм;

E - нормована мінімальна освітленість, Лк; $E = 300$ Лк;

S - площа освітлюваного приміщення (у нашому випадку $S=20\text{м}^2$);

K - коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку $K = 1,5$);

Z - відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку $Z = 1,1$);

n - коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{\text{стін}}$) і стелі ($\rho_{\text{стелі}}$), значення коефіцієнтів дорівнюють $\rho_{\text{стін}} = 50\%$ і $\rho_{\text{стелі}} = 50\%$.

Обчислимо індекс приміщення за формулою:

$$i=S/(h(A+B)), \quad (8.2)$$

де: S - площа приміщення, $S = 20 \text{ м}^2$;

h - розрахункова висота підвісу, $h = 3 \text{ м}$ (співпадає з висотою стелі, т.я. лампи освітлення закріплюються на стелі);

A - ширина приміщення, $A = 4 \text{ м}$;

B - довжина приміщення, $B = 5 \text{ м}$.

Розрахований за формулою індекс приміщення становить $i = 0,74$.

Відповідно до розрахованого значення індексу приміщення в $i = 0,74$ коефіцієнт використання світлового потоку становить $n=0,22$ (з табличних даних коефіцієнтів використання світлового потоку світильників з відповідним типом лампам). Розраховане за формулою значення світлового потоку становить: $F=45000 \text{ Лм}$.

Для розрахунку будемо використовувати світлодіодні панелі LED панель 42Вт 6000K SUNLED 000000127, світловий потік яких $F_n = 3990 \text{ Лм}$.

Кількість панелей визначається за формулою:

$$N=F/F_n, \quad (8.3)$$

де: F - світловий потік,

F_n - світловий потік однієї панелі.

Розрахункова за формулою кількість панелей становить: $N=11,3 \text{ шт}$.

Приймаємо необхідну кількість світлодіодних світильників 12 шт .

8.6 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз умов праці, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

КБПЗ_2023

									Лист
									98
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ				

Література

1. Логістика - Бібліотека підручників та статей Posibniki (2022) - <https://posibniki.com.ua/post-zavdannya-i-meta-logistiki> (дата звернення: 29.09.2023).
2. Методичні вказівки до виконання розрахункового завдання за дисципліною “Комерційна логістика” для студентів денної та заочної форм навчання за спеціальністю: 073 – «Менеджмент» / Укладачі: Гармаш С.В., Віхляєва Н.В. – Харків: НТУ «ХП», 2019. - 58 с.
3. Марченко В.М., Шутюк В.В., Ємцев В.І.. Логістика: Практикум. Навч. посіб. для студ. спеціальності 051 «Економіка» / КПІ ім. Ігоря Сікорського; - Київ: КПІ ім. Ігоря Сікорського, 2021. - 188с.
4. Марченко С. М. Задачник з логістики. - К. : МАУП, 2006. - 68 с.
5. Лебеденко В.С., Кислун О.А.. Огляд методів розв’язання логістичних задач пошуку оптимальних маршрутів / Матеріали VI Міжнародної науково-практичної конференції “Інформаційна безпека та комп’ютерні технології”: тези доповідей, 20-21 квітня 2023р. -Кропивницький: ЦНТУ, 2023. стр.42
6. Яковенко А., Фолькін М.. Система побудови оптимальних маршрутів на основі алгоритмів найкоротших шляхів / Біомедична інженерія і технологія №2 (2019), стр.54-59
7. ANT-LOGISTICS [Електронний ресурс] - Режим доступу: <https://ant-logistics.com/uk/main.html> (дата звернення: 29.09.2023).
8. ANT-LOGISTICS - огляд, відгуки, аналоги [Електронний ресурс] - Режим доступу: <https://www.livebusiness.com.ua/tool/11/> дата звернення: 29.09.2023).

										Лист
										101
Змн.	Лист	№ докум.	Підпис	Дата						

звернення: 29.09.2023).

28. Delphi. Работа с HDD, CD-ROM, ZIP - [Електронний ресурс] - Режим доступу: <http://z-oleg.com/delphi/hardw1.htm> дата звернення: 29.09.2023).

29. Безменов М. І. Основи програмування у середовищі Delphi : навч. посіб. – Харків : НТУ «ХП», 2010. – 608 с.

30. Каплун В. А., Дмитришин О. В., Баришев Ю. В. Захист програмного забезпечення. Частина 2 : навчальний посібник – Вінниця : ВНТУ, 2014. – 105 с.

31. Дудатьєв А., Каплун В., Семеренко В. Захист програмного забезпечення. Ч.1 : навчальний посібник –Вінниця: ВНТУ, 2005. – 140 с.

32. Жилін А. В., Шаповал О. М., Успенський О. А. Технології захисту інформації в інформаційно-телекомунікаційних системах : навч. посіб. – К. : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2021. – 213 с.

33. Остапов С.Е., Євсєєв С. П., Король О. Г. Технології захисту інформації : навчальний посібник – Харків. : Вид. ХНЕУ, 2013. – 476 с.

34. Грайворонський М., Новіков О. Безпека інформаційно-комунікаційних систем. – К.: Видавнича група ВНУ, 2009. – 608с.

35. Вишня В. Б., Гавриш О. С., Рижков Е. В..Основи інформаційної безпеки : навч. посібник -Дніпро: Дніпроп. держ.ун-т внутріш. справ, 2020. 128 с.

36. Захист програмного забезпечення [Електронний ресурс] - Режим доступу: http://ua-referat.com/Захист_програмного_забезпечення дата звернення: 29.09.2023).

37. Кузнецов О. О., Євсєєв С. П., Король О. Г.. Захист інформації в інформаційних системах – Харків. : Вид. ХНЕУ, 2011. – 512 с.

38. Гребенюк А.М., Рибальченко Л.В. Основи управління інформаційною безпекою: навч. посібник -Дніпро: Дніпроп. держ. унт внутріш. справ, 2020. – 144 с.

39. Захист програм від злому [Електронний ресурс] - Режим доступу:

										Лист
										104
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ					

<http://easy-code.com.ua/2011/04/zaxist-program-vid-zlomu/>

40. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. - Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення: 16.10.2022).

41. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

42. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». - Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508> (дата звернення: 16.10.2022).

43. Державні будівельні норми України. Природне і штучне освітлення: ДБН В.2.5-28:2018. - Режим доступу до ресурсу: <https://goo.su/9AkQ> (дата звернення: 16.10.2022).

44. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. - Кропивницький : ЦНТУ, 2022. - 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення 19.09.22).

45. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. - Кропивницький: ЦНТУ, 2022. — 19 с. [Електронний ресурс]. – Режим доступу : <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення: 16.10.2022).

46. Постанова № 42 від 01.12.1999 Головного державного санітарного

										Лист
										105
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ					

лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. - Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99> (дата звернення 19.09.22).

47. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. - Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>

48. Лебеденко В.С., Кислун О.А. Використання класичних методів розв'язання логістичних задач для побудови алгоритму пошуку маршрутів оптимального постачання / Матеріали VII Міжнародної науково-практичної конференції “Інформаційна безпека та комп'ютерні технології” до 30-ти річчя кафедри кібербезпеки та програмного забезпечення: тези доповідей, 1 листопада 2023 р. – Кропивницький: ЦНТУ, 2023. С.74

49. Швачич Г.Г., Овсянніков О.В., Кузьменко В.В., Нечаєва Н.І. Прикладне програмне забезпечення. Навчальний посібник з основ програмування та конструювання прикладного програмного забезпечення в інтегрованому середовищі Delphi – Дніпропетровськ: НМетАУ, 2007. – 58 С.

50. Глинський Я.М., Анохін В.Є., Ряжська В.А. Паскаль. Turbo Pascal і Delphi : Навчальний посібник 8-ме вид. -Львів: “СПІД Глинський” , 2007. -192с.

51. Іванків К.С., Мацьонг Я. Є., Мандзак Т. І. Delphi для початківців Львів: ЛНУ, 2006. -188с.

52. Кащеєв Л.Б., Коваленко С.В., Коваленко С.М. Основи візуального програмування –Харків: Ранок, 2011. -192с

										Лист
										106
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ПЗ					

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування	2
2 Підстава для розробки	2
3 Мета та призначення розробки	2
4 Джерела розробки	2
5 Технічні вимоги.....	2
5.1 Склад продукції.....	2
5.2 Показники призначення	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури	3
5.5 Вимоги до надійності.....	4
5.6 Умови експлуатації	4
5.7 Вимоги до складу та параметрів технічних засобів	4
5.8 Вимоги до інформаційної і програмної сумісності	4
6 Вимоги до програмної документації.....	5
7 Економічні вимоги	5
8 Вимоги щодо охорони праці	6
9 Перелік документів, що розробляються	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	7

					ВКМР-123.23.0080.00.00.ТЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Лебеденко В.С.			Дослідження та програмна реалізація системи логістики оптимальних шляхів доставки товару	Літ.	Арк.	Аркушіє
Перевір.		Кислун О.А.					1	7
Н. Контр.		Коваленко А.С.			ЦНТУ КІ-22МЗ			
Затверд.		Смірнов О.А.						

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи логістики оптимальних шляхів доставки товару.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 36-13 від 04.08.23 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи логістики оптимальних шляхів доставки товару.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література й існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

									Лист
									2
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ТЗ				

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи логістики оптимальних шляхів доставки товару.
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється, повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільш поширені.

									Лист
									3
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ТЗ				

5.5 Вимоги до надійності

Програмні модулі написані за всіма правилами, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні відповідати наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС сімейства Windows і з сумісними з цією платформою пристроями та прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС сімейства Windows.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

									Лист
									4
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ТЗ				

5.8.2 Мова програмування

Середовище Delphi версії 7.

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у вигляді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД.

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

									Лист
									5
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ТЗ				

8 Вимоги щодо охорони праці

У частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз умов праці на робочому місці ІТ-фахівця

9 Перелік документів, що розробляються

- Наукова новизна - 1 аркуш.
- Структурна схема системи - 1 аркуш.
- Функціональна схема системи - 1 аркуш.
- Діаграма процесів - 1 аркуш.
- Блок-схема алгоритму роботи програми - 2 аркуша.
- Показники економічної ефективності - 1 аркуш.
- Пояснювальна записка - 106 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації з теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок-схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

									Лист
									6
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ТЗ				

- 10.6 Віднаходження ПЗ, аналіз отриманих результатів.
- 10.7 Робота над питанням охорони праці і техніки безпеки.
- 10.8 Розрахунок з техніко-економічного обґрунтування.
- 10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 13.12.2023 р.

КБПЗ – 2023

									Лист
									7
Змн.	Лист	№ докум.	Підпис	Дата	ВКРМ-123.23.0080.00.00.ТЗ				

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ
Керівник магістерської роботи
_____ Кислун О.А.

*Дослідження та програмна реалізація системи логістики
оптимальних шляхів доставки товару*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 28

Літера: РП

Кропивницький 2023

```
// ----- SLODT.pas ----- Програмный модуль

unit PROG1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, StdCtrls, Grids, DBGrids, Registry;

type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    N10: TMenuItem;
    N11: TMenuItem;
    N12: TMenuItem;
    N13: TMenuItem;
    N20: TMenuItem;
    N21: TMenuItem;
    N22: TMenuItem;
    N23: TMenuItem;
    N24: TMenuItem;
    N30: TMenuItem;
    N31: TMenuItem;
    N32: TMenuItem;
    N33: TMenuItem;
    N34: TMenuItem;
    N40: TMenuItem;
    N41: TMenuItem;
    N42: TMenuItem;
    N43: TMenuItem;
    N50: TMenuItem;
    N51: TMenuItem;
    N52: TMenuItem;
    N60: TMenuItem;
    N61: TMenuItem;
    Edit1: TEdit;
    StringGrid1: TStringGrid;
    Edit2: TEdit;
    N25: TMenuItem;
    N26: TMenuItem;
    N27: TMenuItem;
    StringGrid2: TStringGrid;
    Label1: TLabel;
    Label2: TLabel;
    OpenFileDialog1: TOpenDialog;
    SaveDialog1: TSaveDialog;
    N35: TMenuItem;
    N36: TMenuItem;
    N37: TMenuItem;
    Edit3: TEdit;
    StringGrid3: TStringGrid;
    Memo1: TMemo;
    Label3: TLabel;
    Button1: TButton;
    Memo2: TMemo;

    procedure N13Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure N21Click(Sender: TObject);
    procedure NotVisible(Sender: TObject);
  end;
end;
```

```

procedure N31Click(Sender: TObject);
procedure Edit1Change(Sender: TObject);
procedure N22Click(Sender: TObject);
procedure MainMenu1Change(Sender: TObject; Source: TMenuItem;
  Rebuild: Boolean);
procedure StringGrid1Click(Sender: TObject);
procedure N23Click(Sender: TObject);
procedure Edit1KeyPress(Sender: TObject; var Key:Char);
procedure N24Click(Sender: TObject);
procedure StringGrid1KeyPress(Sender: TObject; var Key:Char);
procedure StringGrid2Click(Sender: TObject);
procedure StringGrid2KeyPress(Sender: TObject; var Key:Char);
procedure N25Click(Sender: TObject);
procedure N27Click(Sender: TObject);
procedure N26Click(Sender: TObject);
procedure Edit2KeyPress(Sender: TObject; var Key:Char);
procedure Edit2Change(Sender: TObject);
procedure N11Click(Sender: TObject);
procedure N12Click(Sender: TObject);
procedure param(Sender: TObject);
procedure N32Click(Sender: TObject);
procedure N41Click(Sender: TObject);
procedure N33Click(Sender: TObject);
procedure N36Click(Sender: TObject);
procedure N34Click(Sender: TObject);
procedure N35Click(Sender: TObject);
procedure N37Click(Sender: TObject);
procedure N43Click(Sender: TObject);
procedure N42Click(Sender: TObject);
procedure StringGrid3Click(Sender: TObject);
procedure AU(Sender: TObject);
procedure N51Click(Sender: TObject);
procedure N52Click(Sender: TObject);
procedure N61Click(Sender: TObject);
procedure Edit3Change(Sender: TObject);
procedure Button1Click(Sender: TObject);

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  B,pp:array [1..60] of string[255];
  G:ARRAY [1..60,1..60] OF REAL;
  Z,Bk,Dp,PD1,PG2:ARRAY [1..60] OF REAL;
  f,Ri:ARRAY [1..60] OF integer;
  bN,bp,NN:integer;
  p,op:real;
  MamberSertefiket:string[80];
  VolumeSerialNo,snr : DWord;
implementation

{$R *.dfm}

// Блокування вводу букв в кількості пунктів
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
if not (Key in [ '0'..'9',#8,#46])

```

```

then Key := #0;
end;

// Блокування вводу букв в об`ємі потреби
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
if Key='.' then Key:=',';
if (not (Key in [ '0'..'9', ',', '#8,#46]))
or ( (Pos(',',Edit2.text)>0)and (Key=','))
then Key := #0;
end;

// Блокування вводу букв в витратах
procedure TForm1.StringGrid1KeyPress(Sender: TObject; var Key: Char);
begin
if Key='.' then Key:=',';
if (NN=23) or (NN=33) then
if (not (Key in [ '0'..'9', ',', '#8,#46]))
or ((Pos(',', StringGrid1.Cells[2,Stringgrid1.Row])>0)and (Key=',') )
then Key := #0;
if (NN=27) or (NN=37) then
if (not (Key in [ '0'..'9', ',', '#8,#46]))
or ((Pos(',', StringGrid1.Cells[3,Stringgrid1.Row])>0)
and (Key=',') )
then Key := #0;
end;

// Блокування вводу букв в затратах витратах за рейсом
procedure TForm1.StringGrid2KeyPress(Sender: TObject; var Key: Char);
begin
if Key='.' then Key:=',';
if (not (Key in [ '0'..'9', ',', '#8,#46]))
or (((Pos(',', StringGrid2.Cells[Stringgrid2.Col,Stringgrid2.Row])>0) )
and (Key=',') )
then Key := #0;
end;

// виключення-виключення відображення об'єктів
procedure TForm1.NotVisible(Sender: TObject);
var
ii:integer;
begin
//au(Sender);
edit1.Visible:=false;
edit2.Visible:=false;
StringGrid1.Visible:=false;
StringGrid2.Visible:=false;
StringGrid3.Visible:=false;
label2.Visible:=false;
label1.Visible:=false;
label3.Visible:=false;
edit3.Visible:=false;
BUTTON1.Visible:=false;
Memo1.Visible:=false;
Memo2.Visible:=false;
if bn<2 then bn:=0;
if bn>1
then
begin
n21.Caption:='КІЛЬКІСТЬ ПУНКТІВ = '+IntToStr(bn);
n21.Checked:=true;

```

```

n21.Enabled:=false;
n22.Enabled:=true;
n31.Caption:='КІЛЬКІСТЬ ПУНКТИВ = '+IntToStr(bn);
n31.Checked:=true;
n32.Enabled:=true;
n33.Enabled:=true;
n34.Enabled:=true;
n35.Enabled:=true;
n37.Enabled:=true;
end
else
begin
n21.Caption:='КІЛЬКІСТЬ ПУНКТИВ';
n21.Checked:=false;
n21.Enabled:=true;
n31.Checked:=false;
n31.Caption:='КІЛЬКІСТЬ ПУНКТИВ';
n32.Enabled:=false;
n33.Enabled:=false;
n34.Enabled:=false;
n22.Enabled:=false;
n23.Enabled:=false;
n24.Enabled:=false;
n35.Enabled:=false;
n37.Enabled:=false;
end;
if (bp>0) and (bp<=bn)
then
begin
n25.Caption:='ПУНКТ ДОСТАВКИ => '+IntToStr(bp)+'
'+Stringgrid1.Cells[1,Stringgrid1.Row];
n25.Checked:=true;
n25.Enabled:=false;
end
else
begin
bp:=0;
n25.Caption:='ПУНКТ ДОСТАВКИ';
n25.Checked:=false;
end;

if (nn=22) or (nn=32)
or (nn=21) or (nn=31)
or (nn=11)
then
begin
if bn>1
then
begin
n22.Checked:=true;
for ii:=1 to bn do
if n22.Checked then
if (Trim(Stringgrid1.Cells[Stringgrid2.Col,ii])='')
then n22.Checked:=false;
end
else n22.Checked:=false;
if n22.Checked
then
begin
n32.Checked:=true;
n23.Enabled:=true;

```

```

n24.Enabled:=true;
n25.Enabled:=true;
n27.Enabled:=true;
end
else
begin
n32.Checked:=false;
n23.Enabled:=false;
n24.Enabled:=false;
n25.Enabled:=false;
n27.Enabled:=false;
end;
end;
end;

// налаштування при відкритті
procedure TForm1.FormCreate(Sender: TObject);
begin
Form1.caption:='SLODT';
MamberSertefiket:='';
au(Sender);
StringGrid1.Cells[1,0]:='Назва пункту' ;
StringGrid1.Cells[2,0]:='Витрати на пінкті, як прохідному';
StringGrid1.Cells[3,0]:='Наявність';
StringGrid2.Cells[1,0]:='Прибуття';
StringGrid2.Cells[0,1]:='Відправка';
edit1.left:=260;
Edit1.top:=20;
Edit1.Width:=170;
Edit1.Height:=40;
Edit1.Font.Size:=14;
Edit1.Font.Name:='Times New Roman';
Edit1.Font.Style:=[fsBold];
label1.left:=0;
label1.top:=25;
label1.Width:=170;
label1.Height:=40;
label1.Font.Size:=14;
label1.Font.Name:='Times New Roman';
label1.Font.Style:=[fsBold];
label1.Caption:='Кількість пунктів:';
edit2.left:=210;
Edit2.top:=20;
Edit2.Width:=170;
Edit2.Height:=40;
Edit2.Font.Size:=14;
Edit2.Font.Name:='Times New Roman';
Edit2.Font.Style:=[fsBold];
label2.left:=0;
label2.top:=25;
label2.Width:=170;
label2.Height:=40;
label2.Font.Size:=14;
label2.Font.Name:='Times New Roman';
label2.Font.Style:=[fsBold];
label2.Caption:='Об`єм потреби:';
label1.Visible:=false;
edit3.left:=550;
Edit3.text:='';
Edit3.top:=42;
Edit3.Width:=330;

```

```
Edit3.Height:=60;
Edit3.Font.Size:=30;
Edit3.Font.Name:='Times New Roman';
Edit3.Font.Style:=[fsBold];
edit3.Visible:=false;
BUTTON1.left:=55;
BUTTON1.top:=150;
BUTTON1.Width:=800;
BUTTON1.Height:=100;
BUTTON1.Caption:='ВСТАНОВИТИ КОД АКТИВАЦІЇ';
BUTTON1.Font.Style:=[fsBold];
BUTTON1.Font.Name:= 'Arial';
BUTTON1.Font.Size:=16;
NotVisible(Sender);
StringGrid1.Font.Size:=14;
StringGrid1.Font.Name:='Times New Roman';
StringGrid1.Font.Style:=[fsBold];
StringGrid2.Font.Size:=14;
StringGrid2.Font.Name:='Times New Roman';
StringGrid2.Font.Style:=[fsBold];
label3.top:=35;
label3.left:=25;
label3.Width:=170;
label3.Height:=40;
label3.Font.Size:=18;
label3.Font.Name:='Times New Roman';
label3.Font.Style:=[fsBold];
label3.Caption:='ВВЕДІТЬ НОМЕР ЛІЦЕНЗІЇ '+#13+' ДЛЯ АКТИВАЦІЇ ДОБДТКУ';
memo2.ReadOnly:=true;
//кількість пунктів
bN:=0;
bp:=0;
edit2.text:='0';
edit1.text:='0';
  n21.Enabled:=true;
  n22.Checked:=false;
  n22.Enabled:=false;
  n23.Checked:=false;
  n23.Enabled:=false;
  n24.Checked:=false;
  n24.Enabled:=false;
  n25.Checked:=false;
  n25.Enabled:=false;
  n26.Checked:=false;
  n26.Enabled:=true;
  n27.Checked:=false;
  n27.Enabled:=false;
  n31.Enabled:=true;
  n32.Checked:=false;
  n32.Enabled:=false;
  n33.Checked:=false;
  n33.Enabled:=false;
  n34.Checked:=false;
  n34.Enabled:=false;
  n35.Checked:=false;
  n35.Enabled:=false;
  n36.Checked:=false;
  n36.Enabled:=true;
  n37.Checked:=false;
  n37.Enabled:=false;
```

```

end;

//
procedure TForm1.N13Click(Sender: TObject);
begin
  NN:=13;
  NotVisible(Sender);
  Halt;
end;

//введення кількості пунктів
procedure TForm1.N21Click(Sender: TObject);
begin
  NN:=21;
  if bN<2 then bn:=0;
  edit1.text:=IntToStr(bN);;
  edit1.Visible:=true;
  labell1.Visible:=true;
end;

// Редагування кількості пунктів
procedure TForm1.N31Click(Sender: TObject);
begin
  NN:=31;
  edit1.Visible:=true;
  labell1.Visible:=true;
  edit1.text:=IntToStr(bN);;
end;

// Обмеження на кількість пунктів
procedure TForm1.Edit1Change(Sender: TObject);
begin
  bN:=StrToInt('0'+edit1.text);
  if bN>60
  then
  begin
    bN:=60;
    edit1.text:='60';
  end;
end;

// ВВЕДЕННЯ ПУНТІВ
procedure TForm1.N22Click(Sender: TObject);
var
  ii : Integer;
begin
  NN:=22;
  NotVisible(Sender);
  StringGrid1.left:=0;
  StringGrid1.top:=20;
  StringGrid1.Visible:=true;
  StringGrid1.DefaultColWidth:=100;
  StringGrid1.Visible:=true;
  StringGrid1.ColWidths[0] := 40;
  StringGrid1.Width:=form1.Width-50;
  IF form1.Height-165>250
  THEN
  StringGrid1.Height:=form1.Height-165
  ELSE
  StringGrid1.Height:=250;
  StringGrid1.ColCount:=2;

```

```

StringGrid1.DefaultRowHeight:=40;
StringGrid1.ColWidths[1]:=StringGrid1.Width-StringGrid1.ColWidths[0]-10;
StringGrid1.RowCount:=bn+1;
StringGrid1.FixedRows:=1;
StringGrid1.FixedCols:=1;
for ii := 1 to StringGrid1.RowCount-1
  do StringGrid1.Cells[0, ii]:= IntToStr(ii);
end;

// Реакція на активізацію меню
procedure TForm1.MainMenuChange(Sender: TObject; Source: TMenuItem;
  Rebuild: Boolean);
begin
  NotVisible(Sender);
end;

// Фредагування/педагування пунктів
procedure TForm1.StringGrid1Click(Sender: TObject);
begin
Stringgrid1.Options:= Stringgrid1.Options - [goEditing];
if (Trim(Stringgrid1.Cells[1,Stringgrid1.Row]='') AND (nn=22) AND
  (Stringgrid1.Col=1)
  then
Stringgrid1.Options:= Stringgrid1.Options + [goEditing];
if (Trim(Stringgrid1.Cells[2,Stringgrid1.Row]='') AND (nn=23) AND
  (Stringgrid1.Col=2)
  then
Stringgrid1.Options:= Stringgrid1.Options + [goEditing];
if (Trim(Stringgrid1.Cells[3,Stringgrid1.Row]='') AND (nn=27) AND
  (Stringgrid1.Col=3)
  then
Stringgrid1.Options:= Stringgrid1.Options + [goEditing];
if nn=25
  then
  begin
  bp:=Stringgrid1.Row;
  n25.Caption:='ПУНКТ ДОСТАВКИ => '+IntToStr(bp)+'
  '+Stringgrid1.Cells[1,Stringgrid1.Row];
  n35.Caption:='ПУНКТ ДОСТАВКИ => '+IntToStr(bp)+'
  '+Stringgrid1.Cells[1,Stringgrid1.Row];
  end;
if (nn=32) AND (Stringgrid1.Col=1)
  then
Stringgrid1.Options:= Stringgrid1.Options + [goEditing];
if (nn=33) AND (Stringgrid1.Col=2)
  then
Stringgrid1.Options:= Stringgrid1.Options + [goEditing];
if (nn=37) AND (Stringgrid1.Col=3)
  then
Stringgrid1.Options:= Stringgrid1.Options + [goEditing];
if nn=35
  then
  begin
  bp:=Stringgrid1.Row;
  n25.Caption:='ПУНКТ ДОСТАВКИ => '+IntToStr(bp)+'
  '+Stringgrid1.Cells[1,Stringgrid1.Row];
  n35.Caption:='ПУНКТ ДОСТАВКИ => '+IntToStr(bp)+'
  '+Stringgrid1.Cells[1,Stringgrid1.Row];
  end;
end;
end;

```

```

// ЗАТРАТИ НА ЗБЕРІГАННЯ
procedure TForm1.N23Click(Sender: TObject);
var ii:integer;
begin
param(Sender);
NN:=23;
StringGrid1.ColCount:=3;
NotVisible(Sender);
StringGrid1.left:=0;
StringGrid1.top:=20;
StringGrid1.Visible:=true;
StringGrid1.DefaultColWidth:=100;
StringGrid1.Visible:=true;
StringGrid1.Width:=form1.Width-50;
IF form1.Height-165>250
THEN
StringGrid1.Height:=form1.Height-165
ELSE
StringGrid1.Height:=250;
StringGrid1.ColWidths[2]:=500;
StringGrid1.ColWidths[1]:=StringGrid1.Width-StringGrid1.ColWidths[0]-
StringGrid1.ColWidths[2]-40;
StringGrid1.RowCount:=bn+1;
StringGrid1.FixedRows:=1;
StringGrid1.FixedCols:=1;
StringGrid1.ColCount:=3;
StringGrid1.DefaultRowHeight:=40;
    for ii := 1 to StringGrid1.RowCount-1
        do StringGrid1.Cells[0, ii]:= IntToStr(ii);
end;

// Введення затрат на рейс
procedure TForm1.N24Click(Sender: TObject);
var qqq:real;
    ii:integer;
begin
NotVisible(Sender);
NN:=24;
StringGrid2.left:=0;
StringGrid2.top:=20;
StringGrid2.Visible:=true;;
StringGrid2.Width:=form1.Width-50;
StringGrid2.DefaultColWidth:=150;
IF form1.Height-165>250
THEN
StringGrid2.Height:=form1.Height-165
ELSE
StringGrid2.Height:=250;
StringGrid2.RowCount:=bn+2;
StringGrid2.ColCount:=bn+2;
StringGrid2.FixedRows:=2;
StringGrid2.FixedCols:=2;
StringGrid2.DefaultRowHeight:=40;
for ii:= 1 to bn
    do
    begin
StringGrid2.Cells[0, ii+1]:= IntToStr(ii);
StringGrid2.Cells[ii+1, 0]:= IntToStr(ii);
StringGrid2.Cells[1, ii+1]:= StringGrid1.Cells[1, ii];
StringGrid2.Cells[ii+1, 1]:= StringGrid1.Cells[1, ii];
end;

```

```

end;

// Предагування/педагування вартості рейсів
procedure TForm1.StringGrid2Click(Sender: TObject);
begin
Stringgrid2.Options:= Stringgrid2.Options - [goEditing];
if (((Trim(Stringgrid2.Cells[Stringgrid2.Col,Stringgrid2.Row])='')
AND (nn=24)) or (nn=34)) AND (Stringgrid2.Col>1) AND (Stringgrid2.Row>1)
and (not (Stringgrid2.Col=Stringgrid2.Row) )
then
Stringgrid2.Options:= Stringgrid1.Options + [goEditing];
end;

// Вибір пункту доставки
procedure TForm1.N25Click(Sender: TObject);
var
ii : Integer;
begin
param(Sender);
NN:=25;
NotVisible(Sender);
StringGrid1.left:=0;
StringGrid1.top:=20;
StringGrid1.Visible:=true;
StringGrid1.DefaultColWidth:=100;
StringGrid1.Visible:=true;
StringGrid1.ColWidths[0] := 40;
StringGrid1.Width:=form1.Width-50;
IF form1.Height-165>250
THEN
StringGrid1.Height:=form1.Height-165
ELSE
StringGrid1.Height:=250;
StringGrid1.ColCount:=2;
StringGrid1.DefaultRowHeight:=40;
StringGrid1.ColWidths[1]:=StringGrid1.Width-StringGrid1.ColWidths[0]-10;
StringGrid1.RowCount:=nn+1;
StringGrid1.FixedRows:=1;
StringGrid1.FixedCols:=1;
for ii := 1 to StringGrid1.RowCount-1
do StringGrid1.Cells[0, ii]:= IntToStr(ii);
end;

//
procedure TForm1.N27Click(Sender: TObject);
var ii:integer;
begin
param(Sender);
NotVisible(Sender);
NN:=27;
StringGrid1.ColCount:=4;
StringGrid1.left:=0;
StringGrid1.top:=20;
StringGrid1.Visible:=true;
StringGrid1.DefaultColWidth:=100;
StringGrid1.Visible:=true;
StringGrid1.Width:=form1.Width-50;
IF form1.Height-165>250
THEN
StringGrid1.Height:=form1.Height-165
ELSE

```

```

StringGrid1.Height:=250;
StringGrid1.ColWidths[2]:=0;
StringGrid1.ColWidths[3]:=450;
StringGrid1.ColWidths[1]:=StringGrid1.Width-StringGrid1.ColWidths[0]-
StringGrid1.ColWidths[2]-StringGrid1.ColWidths[3]-40;
StringGrid1.RowCount:=bn+1;
StringGrid1.FixedRows:=1;
StringGrid1.FixedCols:=1;
StringGrid1.DefaultRowHeight:=40;
    for ii := 1 to StringGrid1.RowCount-1
        do StringGrid1.Cells[0, ii]:= IntToStr(ii);
end;

// Введення потреб
procedure TForm1.N26Click(Sender: TObject);
begin
param(Sender);
NN:=26;
edit2.Visible:=true;
label2.Visible:=true;
end;

//
procedure TForm1.Edit2Change(Sender: TObject);
begin
p:=StrToFloat('0'+edit2.text);
end;

//
procedure TForm1.N11Click(Sender: TObject);
var
    Fd : TextFile;
    jj,gg:integer;
    C:char;
begin
nn:=11;
if OpenFileDialog1.Execute=true
then
begin
AssignFile(Fd,OpenDialog1.FileName);
ReSet(Fd);
edit1.text:='';
repeat
Read(Fd,c);
if not (c=#13) then edit1.text:=edit1.text+c;
until (c=#13);
bn:=StrToInt(edit1.text);
for jj:=1 to bn do
begin
StringGrid1.Cells[1,jj]:='';
repeat
Read(Fd,c);
if not (c=#13)
then
StringGrid1.Cells[1,jj]:=StringGrid1.Cells[1,jj]+c;
until (c=#13);
end;
for jj:=1 to bn do
begin
StringGrid1.Cells[2,jj]:='';
repeat

```

```

Read(Fd,c);
if ( c in [ '0'..'9',','] ) // not (c=#13)
then
StringGrid1.Cells[2,jj]:=StringGrid1.Cells[2,jj]+c;
until (c=#13);
end;
for jj:=1 to bn do
begin
StringGrid1.Cells[3,jj]:='';
repeat
Read(Fd,c);
if ( c in [ '0'..'9',','] ) // not (c=#13)
then
StringGrid1.Cells[3,jj]:=StringGrid1.Cells[3,jj]+c;
until (c=#13);
end;
edit2.text:='';
repeat
Read(Fd,c);
if ( c in [ '0'..'9',','] ) then edit2.text:=edit2.text+c;
until (c=#13);
for jj:=1 to bn do
for gg:=1 to bn do
begin
StringGrid2.Cells[gg+1,jj+1]:='';
repeat
Read(Fd,c);
if ( c in [ '0'..'9',','] )
then StringGrid2.Cells[gg+1,jj+1]:=StringGrid2.Cells[gg+1,jj+1]+c;
until (c=#13);
end;
CloseFile(Fd);
end;
end;

//
procedure TForm1.N12Click(Sender: TObject);
var
    Fd : TextFile;
    jj,gg:integer;
begin
param(Sender);
nn:=12;
if SaveDialog1.Execute=true
then
begin
AssignFile(Fd,SaveDialog1.FileName);
Rewrite(Fd);
Writeln(Fd,edit1.text);
bn:=StrToInt(edit1.text);
for jj:=1 to bn do
Writeln(Fd,StringGrid1.Cells[1,jj]);
for jj:=1 to bn do
Writeln(Fd,StringGrid1.Cells[2,jj]);
for jj:=1 to bn do
Writeln(Fd,StringGrid1.Cells[3,jj]);
Writeln(Fd,edit2.text);
for jj:=1 to bn do
for gg:=1 to bn do
Writeln(Fd,StringGrid2.Cells[gg+1,jj+1]);
CloseFile(Fd);

```

```

end;
end;

// виключення-виключення відображення об'єктів
procedure TForm1.param(Sender: TObject);
begin
if (nn in [ 21,31])
then
begin
bn:=StrToInt(edit1.text);
if bn<2
then
begin
bn:=0;
n27.Checked:=false;
end;
end;
end;

//
procedure TForm1.N32Click(Sender: TObject);
var
ii : Integer;
begin
NotVisible(Sender);
NN:=32;
StringGrid1.left:=0;
StringGrid1.top:=20;
StringGrid1.Visible:=true;
StringGrid1.DefaultColWidth:=100;
StringGrid1.Visible:=true;
StringGrid1.ColWidths[0] := 40;
StringGrid1.Width:=form1.Width-50;
IF form1.Height-165>250
THEN
StringGrid1.Height:=form1.Height-165
ELSE
StringGrid1.Height:=250;
StringGrid1.ColCount:=2;
StringGrid1.DefaultRowHeight:=40;
StringGrid1.ColWidths[1]:=StringGrid1.Width-StringGrid1.ColWidths[0]-10;
StringGrid1.RowCount:=bn+1;
StringGrid1.FixedRows:=1;
StringGrid1.FixedCols:=1;
for ii := 1 to StringGrid1.RowCount-1
do StringGrid1.Cells[0, ii]:= IntToStr(ii);
end;

//
procedure TForm1.N41Click(Sender: TObject);
VAR
ii, JJ, GG, AA, r: INTEGER;
dpi: real;
begin
NN:=41;
bn:=StrToInt(edit1.text);
FOR JJ:=1 TO bn do
FOR GG:=1 TO bn do
if StrToFloat('0'+StringGrid2.Cells[jj+1,gg+1])>0
then G[gg, jj]:=StrToFloat('0'+StringGrid2.Cells[jj+1,gg+1])
else G[gg, jj]:=-1 ;

```

```

FOR JJ:=1 TO bn do Z[jj]:=StrToFloat(StringGrid1.Cells[2,jj]);
FOR JJ:=1 TO bn do Bk[jj]:=StrToFloat('0'+StringGrid1.Cells[3,jj]);
FOR ii:=1 TO bn do F[ii]:=100;
////////// реалізація алгоритму пошуку оптимального шляху
FOR ii:=1 TO bn do pp[ii]:='';
FOR ii:=1 TO bn do DP[ii]:=0;
pp[bp]:=IntToStr(bp);
F[1]:=bp;
DP[f[1]]:=0;
r:=1;
Ri[F[1]]:=r;
////////// не алгоритм а інтерфейс //////////
StringGrid1.Cells[5,F[1]]:=StringGrid1.Cells[1,F[1]]; //
//////////
FOR ii:=1 TO bn-1 do
  FOR jj:=1 TO bn do
    if (not (F[ii]=JJ)) and (G[jj,F[ii]] >0)
    then
      begin
        dpi:=DP[f[ii]]+G[jj,F[ii]];
        if (not (ii=1))
          then dpi:=dpi+Z[F[ii]];
        if (DP[jj]>dpi) or (pp[jj]='')
          then
            begin
              if (pp[jj]='')
                then
                  begin
                    r:=r+1;
                    Ri[jj]:=r;
                  end;
              F[Ri[jj]]:=jj;
              DP[jj]:=dpi;
              PP[jj]:= IntToStr(jj)+'-'+PP[F[ii]];
            end;
        //////////// не алгоритм а для інтерфейсу //////////
StringGrid1.Cells[5,jj]:= //
StringGrid1.Cells[1,jj]+'=>'+StringGrid1.Cells[5,F[ii]]; //
//////////
        aa:=Ri[jj];
        while dpi<DP[F[aa-1]]
          do begin
            F[aa]:=F[aa-1];
            Ri[F[aa]]:=aa;
            aa:=aa-1;
            F[aa]:=jj;
            Ri[F[aa]]:=aa;
          end;
        end;
      end;
  end;
end;
{
  ЗА РЕЛОКОВАНО ЕЛЕМЕТИ АЛНОРИТМУ,
  ЯК НЕПОТРІВНЕ ДЛЯ ПРОГРАМИ В ДАНІЙ РКАЛІЗАЦІЇ
  ПЕРЕДБАЧАЄТЬСЯ НАЗАЛЕЖНИЙ ВИВІД РЕЗУЛЬТАТУ
  ЧЕРЕЗ ІНТЕРФЕСНІ ЕЛЕМЕНТИ ДЕЛФІ
  op:=p;
  r:=0;
  while (op>0) or (r<bn)
    do begin
      r:=r+1;
      if Bk[F[r]]>op
        then PD1[r]:=op
        else PD1[r]:=Bk[F[r]];
    end;
}

```

```

        PD2[r]:=PP[F[r]];
        op:=op-PD1[r];
        end;
    }
    //////////////////////////////////////
StringGrid3.ColCount:=5;
StringGrid3.RowCount:=bn+1;;
FOR ii:=1 TO bn
    do StringGrid3.Cells[0, ii]:= IntToStr(ii) ;
        StringGrid3.Cells[1, 0]:='База';
        StringGrid3.Cells[2, 0]:='Об`єм';
        StringGrid3.Cells[3, 0]:='Ціна';
        StringGrid3.Cells[4, 0]:='Шлях';
    FOR ii:=1 TO bn do
        begin
            StringGrid3.Cells[1, ii]:=StringGrid1.Cells[1,F[ii]];
            StringGrid3.Cells[2, ii]:=StringGrid1.Cells[3,F[ii]];
            StringGrid3.Cells[3, ii]:=FloatToStr( DP [ F [ ii ] ] );
            StringGrid3.Cells[4, ii]:=StringGrid1.Cells[5,F[ii]];
        end;
StringGrid3.left:=0;
StringGrid3.top:=20;
StringGrid3.Width:=form1.Width-50;
StringGrid3.ColWidths[0]:= 30;
StringGrid3.ColWidths[1]:= 150;
StringGrid3.ColWidths[2]:= 80;
StringGrid3.ColWidths[3]:= 80;
StringGrid3.ColWidths[4]:= form1.Width
-StringGrid3.ColWidths[0]
-StringGrid3.ColWidths[1]
-StringGrid3.ColWidths[2]
-StringGrid3.ColWidths[3]-60;
StringGrid3.Font.Size:=12;
StringGrid3.Font.Name:='Times New Roman';
StringGrid3.Font.Name:='Times New Roman';
StringGrid3.DefaultRowHeight:=30;
StringGrid3.Height:=form1.Height-165;
StringGrid3.Visible:=true;
end;

//
procedure TForm1.N33Click(Sender: TObject);
var ii:integer;
begin
    param(Sender);
    NN:=33;
    StringGrid1.ColCount:=3;
    NotVisible(Sender);
    StringGrid1.left:=0;
    StringGrid1.top:=20;
    StringGrid1.Visible:=true;
    StringGrid1.DefaultColWidth:=100;
    StringGrid1.Visible:=true;
    StringGrid1.Width:=form1.Width-50;
    IF form1.Height-165>250
    THEN
        StringGrid1.Height:=form1.Height-165
    ELSE
        StringGrid1.Height:=250;
    StringGrid1.ColWidths[2]:=500;

```

```

StringGrid1.ColWidths[1]:=StringGrid1.Width-StringGrid1.ColWidths[0]-
StringGrid1.ColWidths[2]-40;
StringGrid1.RowCount:=bn+1;
StringGrid1.FixedRows:=1;
StringGrid1.FixedCols:=1;
StringGrid1.ColCount:=3;
StringGrid1.DefaultRowHeight:=40;
    for ii := 1 to StringGrid1.RowCount-1
        do StringGrid1.Cells[0, ii]:= IntToStr(ii);
end;

//
procedure TForm1.N36Click(Sender: TObject);
begin
param(Sender);
NN:=36;
edit2.Visible:=true;
label2.Visible:=true;
end;

//
procedure TForm1.N34Click(Sender: TObject);
var qqq:real;
    ii:integer;
begin
NotVisible(Sender);
NN:=34;
StringGrid2.left:=0;
StringGrid2.top:=20;
StringGrid2.Visible:=true;;
StringGrid2.Width:=form1.Width-50;
StringGrid2.DefaultColWidth:=150;
IF form1.Height-165>250
THEN
StringGrid2.Height:=form1.Height-165
ELSE
StringGrid2.Height:=250;
StringGrid2.RowCount:=bn+2;
StringGrid2.ColCount:=bn+2;
StringGrid2.FixedRows:=2;
StringGrid2.FixedCols:=2;
StringGrid2.DefaultRowHeight:=40;
for ii:= 1 to bn
do
begin
StringGrid2.Cells[0, ii+1]:= IntToStr(ii);
StringGrid2.Cells[ii+1, 0]:= IntToStr(ii);
StringGrid2.Cells[1, ii+1]:= StringGrid1.Cells[1, ii];
StringGrid2.Cells[ii+1, 1]:= StringGrid1.Cells[1, ii];
end;
end;

//
procedure TForm1.N35Click(Sender: TObject);
var
    ii : Integer;
begin
param(Sender);
NN:=35;
NotVisible(Sender);
StringGrid1.left:=0;

```

```

StringGrid1.top:=20;
StringGrid1.Visible:=true;
StringGrid1.DefaultColWidth:=100;
StringGrid1.Visible:=true;
StringGrid1.ColWidths[0] := 40;
StringGrid1.Width:=form1.Width-50;
IF form1.Height-165>250
THEN
StringGrid1.Height:=form1.Height-165
ELSE
StringGrid1.Height:=250;
StringGrid1.ColCount:=2;
StringGrid1.DefaultRowHeight:=40;
StringGrid1.ColWidths[1]:=StringGrid1.Width-StringGrid1.ColWidths[0]-10;
StringGrid1.RowCount:=bn+1;
StringGrid1.FixedRows:=1;
StringGrid1.FixedCols:=1;
for ii := 1 to StringGrid1.RowCount-1
do StringGrid1.Cells[0, ii]:= IntToStr(ii);
end;

//
procedure TForm1.N37Click(Sender: TObject);
var ii:integer;
begin
param(Sender);
NotVisible(Sender);
NN:=37;
StringGrid1.ColCount:=4;
StringGrid1.left:=0;
StringGrid1.top:=20;
StringGrid1.Visible:=true;
StringGrid1.DefaultColWidth:=100;
StringGrid1.Visible:=true;
StringGrid1.Width:=form1.Width-50;
IF form1.Height-165>250
THEN
StringGrid1.Height:=form1.Height-165
ELSE
StringGrid1.Height:=250;
StringGrid1.ColWidths[2]:=0;
StringGrid1.ColWidths[3]:=450;
//StringGrid1.ColWidths[4]:=0;
StringGrid1.ColWidths[1]:=StringGrid1.Width-StringGrid1.ColWidths[0]-
StringGrid1.ColWidths[2]-StringGrid1.ColWidths[3]-40;
StringGrid1.RowCount:=bn+1;
StringGrid1.FixedRows:=1;
StringGrid1.FixedCols:=1;
StringGrid1.DefaultRowHeight:=40;
for ii := 1 to StringGrid1.RowCount-1
do StringGrid1.Cells[0, ii]:= IntToStr(ii);
end;

procedure TForm1.N43Click(Sender: TObject);
VAR
ii, JJ, GG, AA, r: INTEGER;
dpi, BCOGO :real;
begin
NN:=43;
bn:=StrToInt(edit1.text);
FOR JJ:=1 TO bn do

```

```

FOR GG:=1 TO bn do
  if StrToFloat('0'+StringGrid2.Cells[jj+1,gg+1])>0
    then G[gg,jj]:=StrToFloat('0'+StringGrid2.Cells[jj+1,gg+1])
    else G[gg,jj]:=-1 ;
FOR JJ:=1 TO bn do Z[jj]:=StrToFloat(StringGrid1.Cells[2,jj]);
FOR JJ:=1 TO bn do Bk[jj]:=StrToFloat('0'+StringGrid1.Cells[3,jj]);
FOR ii:=1 TO bn do F[ii]:=100;
////////// реалізація алгоритму пошуку оптимального шляху
FOR ii:=1 TO bn do pp[ii]:='';
FOR ii:=1 TO bn do DP[ii]:=0;
pp[bp]:=IntToStr(bp);
F[1]:=bp;
DP[f[1]]:=0;
r:=1;
Ri[F[1]]:=r;
////////// не алгоритм а інтерфейс //////////
StringGrid1.Cells[5,F[1]]:=StringGrid1.Cells[1,F[1]]; //
//////////
FOR ii:=1 TO bn-1 do
  FOR jj:=1 TO bn do
    if (not (F[ii]=JJ)) and (G[jj,F[ii]] >0)
      then
        begin
          dpi:=DP[f[ii]]+G[jj,F[ii]];
          if (not (ii=1))
            then dpi:=dpi+Z[F[ii]];
          if (DP[jj]>dpi) or (pp[jj]='')
            then
              begin
                if (pp[jj]='')
                  then
                    begin
                      r:=r+1;
                      Ri[jj]:=r;
                    end;
                F[Ri[jj]]:=jj;
                DP[jj]:=dpi;
                PP[jj]:= IntToStr(jj)+'-'+PP[F[ii]];
////////// не алгоритм а для інтерфейсу //////////
StringGrid1.Cells[5,jj]:= //
StringGrid1.Cells[1,jj]+'>'+StringGrid1.Cells[5,F[ii]]; //
//////////
                aa:=Ri[jj];
                while dpi<DP[F[aa-1]]
                  do begin
                    F[aa]:=F[aa-1];
                    Ri[F[aa]]:=aa;
                    aa:=aa-1;
                    F[aa]:=jj;
                    Ri[F[aa]]:=aa;
                  end;
              end;
            end;
          end;
//////////
StringGrid3.ColCount:=5;
StringGrid3.RowCount:=bn+1;;
FOR ii:=1 TO bn
  do StringGrid3.Cells[0, ii]:= IntToStr(ii) ;
  StringGrid3.Cells[1, 0]:='База';
  StringGrid3.Cells[2, 0]:='Об`ем';
  StringGrid3.Cells[3, 0]:='Ціна';

```

```

StringGrid3.Cells[4, 0]:='Шлях';
FOR ii:=1 TO bn do
begin
StringGrid3.Cells[1, ii]:=StringGrid1.Cells[1,F[ii]];
StringGrid3.Cells[2, ii]:=StringGrid1.Cells[3,F[ii]];
StringGrid3.Cells[3, ii]:=FloatToStr( DP [ F [ ii ] ] );
StringGrid3.Cells[4, ii]:=StringGrid1.Cells[5,F[ii]];
end;
StringGrid3.left:=0;
StringGrid3.top:=20;
StringGrid3.Width:=form1.Width-50;
StringGrid3.ColWidths[0]:= 30;
StringGrid3.ColWidths[1]:= 150;
StringGrid3.ColWidths[2]:= 80;
StringGrid3.ColWidths[3]:= 80;
StringGrid3.ColWidths[4]:= form1.Width-StringGrid3.ColWidths[0]
-StringGrid3.ColWidths[1]-StringGrid3.ColWidths[2]
-StringGrid3.ColWidths[3]-60;
StringGrid3.Font.Size:=12;
StringGrid3.Font.Name:='Times New Roman';
StringGrid3.Font.Name:='Times New Roman';
StringGrid3.DefaultRowHeight:=30;
StringGrid3.Height:=form1.Height-165;
StringGrid3.Visible:=true;
BCOGO:=0;
op:=p;
ii:=1;
jj:=1;
while (op>0) and not(ii>bn) do
begin
if StrToFloat(StringGrid3.Cells[2,ii])>0
then
begin
StringGrid3.Cells[1, jj]:=StringGrid3.Cells[1, ii];
if (op-StrToFloat(StringGrid3.Cells[2, ii])<0)
then
begin
StringGrid3.Cells[2, jj]:=FloatToStr(op);
BCOGO:=BCOGO+op;
op:=0;
end
else
begin
StringGrid3.Cells[2, jj]:=StringGrid3.Cells[2, ii];
BCOGO:=BCOGO+StrToFloat(StringGrid3.Cells[2, ii]);
op:=op-StrToFloat(StringGrid3.Cells[2, ii]);
end;
StringGrid3.Cells[3, jj]:=StringGrid3.Cells[3, ii];
StringGrid3.Cells[4, jj]:=StringGrid3.Cells[4, ii];
jj:=jj+1;
end;
ii:=ii+1;
end;
StringGrid3.Cells[0, jj]:='';
StringGrid3.Cells[1, jj]:='Всього: ';
StringGrid3.Cells[2, jj]:=FloatToStr(BCOGO);
StringGrid3.Cells[3, jj]:='';
if op>0
then
StringGrid3.Cells[4, jj]:='НЕДОСТАЧА - '+FloatToStr(op)
else

```

```

StringGrid3.Cells[4, jj]:='ПОТРЕБА ЗДІЙСНЕННА';
jj:=jj+1;
StringGrid3.RowCount:=jj;;
while not (jj>bn) do
begin
StringGrid3.Cells[0, jj]:='';
StringGrid3.Cells[1, jj]:='';
StringGrid3.Cells[2, jj]:='';
StringGrid3.Cells[3, jj]:='';
StringGrid3.Cells[4, jj]:='';
jj:=jj+1;
end;
end;

//
procedure TForm1.N42Click(Sender: TObject);
VAR
ii, JJ, GG, AA, r: INTEGER;
dpi: real;
begin
NN:=42;
StringGrid3.ColWidths[1]:= 0;
StringGrid3.ColWidths[2]:= 0;
bn:=StrToInt(edit1.text);
FOR JJ:=1 TO bn do
FOR GG:=1 TO bn do
if StrToFloat('0'+StringGrid2.Cells[jj+1,gg+1])>0
then G[gg, jj]:=StrToFloat('0'+StringGrid2.Cells[jj+1,gg+1])
else G[gg, jj]:=-1 ;
FOR JJ:=1 TO bn do Z[jj]:=StrToFloat(StringGrid1.Cells[2, jj]);
FOR JJ:=1 TO bn do Bk[jj]:=StrToFloat('0'+StringGrid1.Cells[3, jj]);
FOR ii:=1 TO bn do F[ii]:=100;
//////////////////// реалізація алгоритму пошуку оптимального шляху
FOR ii:=1 TO bn do pp[ii]:='';
FOR ii:=1 TO bn do DP[ii]:=0;
pp[bp]:=IntToStr(bp);
F[1]:=bp;
DP[f[1]]:=0;
r:=1;
Ri[F[1]]:=r;
//////////////////// не алгоритм а інтерфейс //////////////////
StringGrid1.Cells[5, F[1]]:=StringGrid1.Cells[1, F[1]]; //
StringGrid1.Cells[6, F[1]]:=StringGrid1.Cells[0, F[1]]; //
////////////////////
FOR ii:=1 TO bn-1 do
FOR jj:=1 TO bn do
if (not (F[ii]=JJ)) and (G[jj, F[ii]] >0)
then
begin
dpi:=DP[f[ii]]+G[jj, F[ii]];
if (not (ii=1))
then dpi:=dpi+Z[F[ii]];
if (DP[jj]>dpi) or (pp[jj]='')
then
begin
if (pp[jj]='')
then
begin
r:=r+1;
Ri[jj]:=r;
end;
end;
end;
end;

```

```

        F[Ri[jj]]:=jj;
        DP[jj]:=dpi;
        PP[jj]:= IntToStr(jj)+'-'+PP[F[ii]];
////////// не алгоритм а для інтерфейсу //////////
StringGrid1.Cells[5,jj]:= //
StringGrid1.Cells[1,jj]+'>'+StringGrid1.Cells[5,F[ii]]; //
StringGrid1.Cells[6,jj]:= //
StringGrid1.Cells[0,jj]+'>'+StringGrid1.Cells[6,F[ii]]; //
//////////
        aa:=Ri[jj];
        while dpi<DP[F[aa-1]]
        do begin
            F[aa]:=F[aa-1];
            Ri[F[aa]]:=aa;
            aa:=aa-1;
            F[aa]:=jj;
            Ri[F[aa]]:=aa;
        end;

    end;

end;

//////////
StringGrid3.ColCount:=6;
StringGrid3.RowCount:=bn+1;;
FOR ii:=1 TO bn
do StringGrid3.Cells[0, ii]:= IntToStr(ii) ;
StringGrid3.Cells[1, 0]:='База';
StringGrid3.Cells[2, 0]:='Об`єм';
StringGrid3.Cells[3, 0]:='Ціна';
StringGrid3.Cells[4, 0]:='Шлях';
FOR ii:=1 TO bn do
begin
StringGrid3.Cells[1, ii]:=StringGrid1.Cells[1,F[ii]];
StringGrid3.Cells[2, ii]:=StringGrid1.Cells[3,F[ii]];
StringGrid3.Cells[3, ii]:=FloatToStr( DP [ F [ ii ] ] );
StringGrid3.Cells[4, ii]:=StringGrid1.Cells[5,F[ii]];
StringGrid3.Cells[1, ii]:=StringGrid1.Cells[6,F[ii]];

end;
StringGrid3.left:=0;
StringGrid3.top:=20;
StringGrid3.Width:=form1.Width-50;
StringGrid3.ColWidths[0]:= 30;
StringGrid3.ColWidths[1]:= 0;
StringGrid3.ColWidths[2]:= 0;
StringGrid3.ColWidths[3]:= 80;
StringGrid3.ColWidths[4]:= form1.Width
-StringGrid3.ColWidths[0]
-StringGrid3.ColWidths[1]
-StringGrid3.ColWidths[2]
-StringGrid3.ColWidths[3]-60;
StringGrid3.Font.Size:=12;
StringGrid3.Font.Name:='Times New Roman';
StringGrid3.Font.Name:='Times New Roman';
StringGrid3.DefaultRowHeight:=30;
StringGrid3.Height:=form1.Height-165;
StringGrid3.Visible:=true;
end;

////
procedure TForm1.StringGrid3Click(Sender: TObject);

```

```

VAR
  JJ,r,tt,ss,sss:INTEGER;
begin
  IF NN=42
  THEN
  BEGIN
  nn:=0;
  TT:=Stringgrid3.Row;
    StringGrid3.Cells[3, 0]:=StringGrid3.Cells[3,tt];
    StringGrid3.Cells[4, 0]:=StringGrid3.Cells[4,tt];
    StringGrid3.Cells[1, 0]:=StringGrid3.Cells[1,tt];
    FOR JJ:=1 TO bn do
  begin
    StringGrid3.Cells[1, jj]:='';
    StringGrid3.Cells[2, jj]:='';
    StringGrid3.Cells[3, jj]:='';
    StringGrid3.Cells[4, jj]:='';
  end;
  if (AnsiPos('=',StringGrid3.Cells[4, 0])>0)
    and
    (AnsiPos('=',(Copy(StringGrid3.Cells[4, 0],
    AnsiPos('=',StringGrid3.Cells[4, 0])+1,
    AnsiPos('*', (StringGrid3.Cells[4, 0]+'*')))))>0)
  then
  begin
  r:=1;
  StringGrid3.Cells[4, r]:=StringGrid3.Cells[4, r-1];
  while AnsiPos('=',StringGrid3.Cells[4, r])>0 do
  begin
  StringGrid3.Cells[4,r+1]:=StringGrid3.Cells[4, r];
  ss :=AnsiPos('=',StringGrid3.Cells[4, r]+'=');
  sss:=AnsiPos('*',StringGrid3.Cells[4, r+1]+'*');
  StringGrid3.Cells[4, r]:=Copy(StringGrid3.Cells[4, r],1,ss-1);
  StringGrid3.Cells[4, r+1]:=Copy(StringGrid3.Cells[4, r+1],ss+2,sss);
  r:=r+1;
  end;
  jj:=r;
  if r>2
  then
  begin
  StringGrid3.Cells[5,1]:=StringGrid3.Cells[4,1];
  StringGrid3.Cells[5,2*r-3]:=StringGrid3.Cells[4,r];
  StringGrid3.Cells[5,2*r-2]:=StringGrid3.Cells[4,r];
  while r>2
  do
  begin
  r:=r-1;
  StringGrid3.Cells[5,2*r-1]:=StringGrid3.Cells[4,r];
  StringGrid3.Cells[5,2*r-2]:=StringGrid3.Cells[4,r];
  end;
  FOR r:=1 TO jj-1
  do
  begin
  StringGrid3.Cells[4,r*2-1]:=StringGrid3.Cells[5,r*2-
  1]+'=>'+StringGrid3.Cells[5,r*2];
  StringGrid3.Cells[5,r*2-1]:='';
  if not (r=jj-1) then StringGrid3.Cells[4,r*2]:=StringGrid3.Cells[5,r*2];
  StringGrid3.Cells[5,r*2]:='';
  end;
  end;
end;

```

```

r:=1;
StringGrid3.Cells[1, r]:=StringGrid3.Cells[1, r-1];
while AnsiPos('=',StringGrid3.Cells[1, r])>0
  do
  begin
  StringGrid3.Cells[1,r+1]:=StringGrid3.Cells[1,r];
  ss :=AnsiPos('=',StringGrid3.Cells[1, r]+'=');
  sss:=AnsiPos('*',StringGrid3.Cells[1, r+1]+'*');
  StringGrid3.Cells[1, r]:=Copy(StringGrid3.Cells[1, r],1,ss-1);
  StringGrid3.Cells[1, r+1]:=Copy(StringGrid3.Cells[1, r+1],ss+2,sss);
  r:=r+1;
  end;
jj:=r;
if r>2
  then
  begin
  StringGrid3.Cells[2,1]:=StringGrid3.Cells[1,1];
  StringGrid3.Cells[2,2*r-3]:=StringGrid3.Cells[1,r];
  StringGrid3.Cells[2,2*r-2]:=StringGrid3.Cells[1,r];
  while r>2
    do
    begin
    r:=r-1;
    StringGrid3.Cells[2,2*r-1]:=StringGrid3.Cells[1,r];
    StringGrid3.Cells[2,2*r-2]:=StringGrid3.Cells[1,r];
    end;
  FOR r:=1 TO jj-1
    do
    begin
    StringGrid3.Cells[3,r*2-
1]:=StringGrid2.Cells[strToint(StringGrid3.Cells[2,r*2])+1,strToint(StringGri
d3.Cells[2,r*2-1])+1];
    StringGrid3.Cells[1,r*2-1]:=StringGrid3.Cells[2,r*2-
1]+'='+StringGrid3.Cells[2,r*2];
    StringGrid3.Cells[2,r*2-1]='';
    if not (r=jj-1)
      then
      begin
      StringGrid3.Cells[3,r*2]:=
StringGrid1.Cells[2,strToint(StringGrid3.Cells[2,r*2])];
      StringGrid3.Cells[1,r*2]:=StringGrid3.Cells[2,r*2];
      end;
    end;
  end;
  FOR r:=(jj-1)*2 TO bn do StringGrid3.Cells[0,r]:='';
  StringGrid3.rowCount:=(jj-1)*2;
end
else
begin
  StringGrid3.Cells[3, 1]:=StringGrid3.Cells[3,0];
  StringGrid3.Cells[4, 1]:=StringGrid3.Cells[4,0];
  FOR r:=2 TO bn do StringGrid3.Cells[0,r]:='';
  StringGrid3.rowCount:=2;
end;
StringGrid3.ColCount:=5;
StringGrid3.ColWidths[2]:= 0;
StringGrid3.ColWidths[1]:= 0;
StringGrid3.ColWidths[3]:= 80;
StringGrid3.ColWidths[4]:= form1.Width
-StringGrid3.ColWidths[0]
-StringGrid3.ColWidths[1]

```

```

    -StringGrid3.ColWidths[2]
    -StringGrid3.ColWidths[3]-60;
end;
END;

procedure TForm1.AU(Sender: TObject);
var
  VolumeName,FileSystemName : array [0..MAX_PATH-1] of Char;
  MaxComponentLength,FileSystemFlags: Cardinal;
  reg:TRegistry;
  RDate:tDate;
  j:integer;
begin
  // ЧИТАННЯ НОМРА ПРИСТОЮ 'C:\'
  GetVolumeInformation('C:\',VolumeName,MAX_PATH,@VolumeSerialNo,
  MaxComponentLength,FileSystemFlags, FileSystemName,MAX_PATH);
  // ЧИТАННЯ ДАНИХ ЗАПИСАНИХ ДО ПК
  reg:=TRegistry.Create;
  reg.RootKey:=HKEY_LOCAL_MACHINE;
  reg.OpenKey('HardWare', true);
  // ДОСТУП ДО РЕЄСТРУ ЗАМАРКОВАНО
  // В РЕЛІЗІ ПРОГРАМИ
  //РЕАЛІЗОВАНА ТИМЧАСОВЕ
  // ЗБЕРЕЖЕННЯ ДАНИХ ЯК АЛЬТЕРНАТИВА
  // snr:=REG.readInteger('SN');
  if reg.OpenKey('HardWare/data', true)
  then
  begin
    reg.writeDate('HardWare/data',date);
    reg.writeInteger('HardWare/K',1);
  end;
  if
    // ПРОВІРКА ДАНИХ ЗАПИСАНИХ НА ДИСК
    // ЯК ПОСТІЙНОГО ПІДТВЕРДЖЕННЯ АЛІЦЕНЗІЇ
    (MamberSertefiket=INTTOSTR(VolumeSerialNo))
    OR
    // ТИМЧАСОВИЙ СТАТУС ЛІЦЕНЗІЇ
    // БЕЗ ЗАНЕСЕННЯ ДАНИХ ДО ПК
    (EDIT3.TEXT=INTTOSTR(VolumeSerialNo))
  then
  begin
    n10.Enabled:=true;
    n20.Enabled:=true;
    n30.Enabled:=true;
    n40.Enabled:=true;
    n52.Enabled:=false;
    n52.Checked:=true;
    Form1.caption:='S L O D T';
  end
  else
  begin
    n10.Enabled:=false;
    n20.Enabled:=false;
    n30.Enabled:=false;
    n40.Enabled:=false;
    n51.Enabled:=true;
    n52.Enabled:=true;
    n52.Checked:=false;
    Form1.caption:='Slotd <= НЕЗАРЕЄСТОВАНА КОПІЯ ПРОГРАМИ =>';
  end;
end;

```

```

end;

//
procedure TForm1.N51Click(Sender: TObject);
begin
  AU(Sender);
  Memo1.Lines[0]:='  ';
  Memo1.Lines.Add(' ');
  Memo1.Lines.Add('                ЛІЦЕНЗІЙНИЙ ДОГОВІР №
'+INTTOSTR(VolumeSerialNo));
  Memo1.Lines.Add(' ');
  Memo1.Lines.Add(' м. Кропивницький                " __ " _____ 2023 року');
  Memo1.Lines.Add(' ');
  Memo1.Lines.Add(' Магістрант в особі Лебеденка В.С., що діє на підставі ');
  Memo1.Lines.Add(' _____ ');
  Memo1.Lines.Add(' (надалі - "Ліцензіар"), з однієї сторони, ');
  Memo1.Lines.Add(' та _____, ');
  Memo1.Lines.Add(' в особі _____, ');
  Memo1.Lines.Add(' щ о діє на підставі _____ ');
  Memo1.Lines.Add(' з іншої сторони, надалі разом "Сторони", а кожна окремо');
  Memo1.Lines.Add(' "Сторона", враховуючи те, що Ліцензіар володіє виключними');
  Memo1.Lines.Add(' авторськими майновими правами на програмне забезпечення, ');
  Memo1.Lines.Add(' яке визначене в п.1.1. цього Договору, уклали цей ');
  Memo1.Lines.Add(' Ліцензійний договір (надалі - "Договір") про наступне:');
  Memo1.Lines.Add(' ');
  Memo1.Lines.Add(' 1. Предмет Договору ');
  Memo1.Lines.Add(' 1.1.Відповідно до умов, визначених цим Договором,');
  Memo1.Lines.Add(' Ліцензіар надає Ліцензіату невиключну та таку, що не .');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' 2. Розрахунок за договором ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ');
  Memo1.Lines.Add(' 2.1. Вартість користування рограмним забезпеченням ');
  Memo1.Lines.Add(' (ліцензійна п лата), що підлягає сплаті Ліцензіатом');
  Memo1.Lines.Add(' Ліцензіару за весь період її безкоштовної Ліцензії');
  Memo1.Lines.Add(' відсутня, у разі ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' 3. Умови використання, поширення Програмного забезпечення');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ');
  Memo1.Lines.Add(' 4. Порядок здійснення абонентського обслуговування та');
  Memo1.Lines.Add(' інформаційно-технологічного супроводу (ІТС)..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' 5. Права та обов`язки сторін ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' 6. Відповідальність сторін за порушення умов договору ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');
  Memo1.Lines.Add(' ..... ');

```

```

Memo1.Lines.Add(' 7. Вирішення спорів ..... ');
Memo1.Lines.Add(' ..... ');
Memo1.Lines.Add(' ..... ');
Memo1.Lines.Add(' ..... ');
Memo1.Lines.Add(' 8. Строк дії Договору ..... ');
Memo1.Lines.Add(' ..... ');
Memo1.Lines.Add(' ..... ');
Memo1.Lines.Add(' ..... ');
Memo1.Lines.Add(' 9. Інші умови ..... ');
Memo1.Lines.Add(' ..... ');
Memo1.Lines.Add(' ..... ');
Memo1.Lines.Add(' ..... ');
Memo1.Lines.Add(' 10. Юридичні адреси, реквізити та підписи Сторін ');
Memo1.Lines.Add(' ..... ');
Memo1.Lines.Add(' ..... ');
Memo1.Lines.Add(' ..... ');
Memo1.Lines.Add(' ..... ');
Memo1.Lines[0]:= 'Типова угода на використання ПЗ';
Memo1.Width:= form1.Width-70;
Memo1.left:= 25;
Memo1.top:= 25;
Memo1.Font.Size:=20;
Memo1.Font.Name:='Arial';
Memo1.Height:=form1.Height-60;;
Memo1.Visible:=true;
end;

procedure TForm1.N52Click(Sender: TObject);
begin
NN:=52;
NotVisible(Sender);
label3.Visible:=true;
EDIT3.Visible:=true;
BUTTON1.Visible:=TRUE;
end;

procedure TForm1.N61Click(Sender: TObject);
begin
NN:=61;
AU(Sender);
NotVisible(Sender);
Memo2.Lines[0]:= 'Програмний комплекс SLODT розроблено в 2023 році. ';
Memo2.Lines.Add(' Автор розробки Лебеденко В.С. ');
Memo2.Lines.Add(' ');
Memo2.Lines.Add(' ');
Memo2.Lines.Add('Програмний комплекс SLODT призначений для розрахунку
оптимального');
Memo2.Lines.Add('плану доставки товару та являє собою окремих незалежний
програмний');
Memo2.Lines.Add('продукт, який було розроблено в рамках випускної
кваліфікаційної');
Memo2.Lines.Add('роботи за другим (магістерським) рівнем вищої
освіти на тему:');
Memo2.Lines.Add('"Дослідження та програмна реалізація системи логістики
оптимальних');
Memo2.Lines.Add('шляхів доставки товару" магістрантом групи KI-22МЗ ОПП
"Комп`ютерна');
Memo2.Lines.Add('інженерія" спеціальності 123 "Комп`ютерна інженерія"
Лебеденком В.С. ');
Memo2.Lines.Add(' ');
Memo2.Lines.Add(' ');

```

```

Memo2.Lines.Add('Розроблене проектне рішення розроблялося з метою
наочного');
Memo2.Lines.Add('підтвердження отриманих в процесі дослідження
результатів.');
```

```

end;
```

```

procedure TForm1.Edit3Change(Sender: TObject);
begin
AU(Sender);
end;
```

```

procedure TForm1.Button1Click(Sender: TObject);
var
  VolumeName, FileSystemName : array [0..MAX_PATH-1] of Char;
  MaxComponentLength, FileSystemFlags: Cardinal;
  reg:TRegistry;
begin
  GetVolumeInformation('C:\', VolumeName, MAX_PATH, @VolumeSerialNo,
MaxComponentLength, FileSystemFlags, FileSystemName, MAX_PATH);
  reg:=TRegistry.Create;
  reg.RootKey:=HKEY_LOCAL_MACHINE;
  reg.OpenKey('Hardware', true);
  // ДОСТУП ДО РЕЄСТРУ ЗАМАРКОВАНО
  // В РЕЛІЗІ ПРОГРАМИ
  //РЕАЛІЗОВАНА ТИМЧАСОВЕ
  // ЗБЕРЕЖЕННЯ ДАНИХ ЯК АЛЬТЕРНАТИВА
  // REG.WriteInteger('SN', VolumeSerialNo);
  reg.CloseKey;
  reg.Free;
end;
```

```

end.
```