

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації і управління

До захисту допущено:

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ
(підпис) (вл.ім'я, прізвище)

“ ___ ” _____ 2020 р.

Дипломний проєкт
на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
спеціальності 122 «Комп'ютерні науки та інформаційні технології»

на тему: «*Автоматизована система управління технічним
обслуговуванням виробничого обладнання*»

Виконав: студент IV курсу, групи ІС-зб1

_____ Оробей Едуард Віталійович
(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник

_____ Ст. викл. Новікова Поліна Анатоліївна

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

_____ (підпис)

**Консультант з
графічної
документації**

_____ доц., к.т.н., доц. Телішева Тамара Олексіївна

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

_____ (підпис)

Рецензент

_____ доц. каф. ТК, ФІОТ Корнага Ярослав Ігорович

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

_____ (підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____
(підпис)

Київ – 2020 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ
(підпис) (вл.ім'я, прізвище)

“ ” 2020 р.

ЗАВДАННЯ

на дипломний проєкт студенту

Оробєю Едуарду Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема проєкту «Автоматизована система управління технічним
обслуговуванням виробничого обладнання»

керівник проєкту ст. викл. Новікова Поліна Анатоліївна
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом по університету від “7” травня 2020 р. №1081-с

2. Термін подання студентом проєкту “01” червня 2020 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. *Схема структурна варіантів використань*

2. *Схема структурна діяльності*

3. *Схема бази даних*

4. *Схема структурна класів програмного забезпечення*

5. *Рішення з математичного забезпечення*

6. *Креслення вигляду екранних форм*

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «13» квітня 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>20.04.2020</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>25.04.2020</i>	
3.	<i>Постановка та формалізація задачі</i>	<i>26.04.2020</i>	
4.	<i>Розробка інформаційного забезпечення</i>	<i>1.05.2020</i>	
5.	<i>Алгоритмізація задачі</i>	<i>3.05.2020</i>	
6.	<i>Обґрунтування використовуваних технічних засобів</i>	<i>5.05.2020</i>	
7.	<i>Розробка програмного забезпечення</i>	<i>7.05.2020</i>	
8.	<i>Налагодження програми</i>	<i>8.05.2020</i>	
9.	<i>Виконання графічних документів</i>	<i>10.05.2020</i>	
10.	<i>Оформлення пояснювальної записки</i>	<i>13.05.2020</i>	
11.	<i>Подання ДП на попередній захист</i>	<i>15.05.2020</i>	
12.	<i>Подання ДП на основний захист</i>	<i>01.06.2020</i>	
13.	<i>Подання ДП рецензенту</i>	<i>02.06.2020</i>	

Студент

Едуард ОРОБСЬ

Керівник

Поліна НОВІКОВА

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації і управління

УЗГОДЖЕНО

Керівник проєкту

_____ Поліна НОВІКОВА

(підпис)

(вл. ім'я, прізвище)

“13” квітня 2020 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ

(підпис)

(вл. ім'я, прізвище)

“14” квітня 2020 р.

Автоматизована система управління технічним обслуговуванням
Виробничого обладнання

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр *ДП 6106.01.000 ТЗ*

на 9 сторінках

Київ – 2020 року

ЗМІСТ

1	ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	3
1.1	Повне найменування системи та її умовне позначення.....	3
1.2	Найменування організації-замовника та організацій-учасників робіт.....	3
1.3	Перелік документів, на підставі яких створюється система (Завдання на ДП).....	3
1.4	Планові терміни початку і закінчення роботи зі створення системи.....	3
2	ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ	4
2.1	Призначення системи.....	4
2.2	Цілі створення системи.....	4
3	ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	6
4.1	Вимоги до функціональних характеристик.....	6
4.2	Вимоги до надійності.....	6
4.3	Умови експлуатації (тільки для систем, специфіка яких передбачає особливі умови експлуатації).....	6
4.4	Вимоги до складу і параметрів технічних засобів.....	6
5	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	8
6	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	9
6.1	Види випробувань.....	9

					ДП 6106.01.000 ТЗ			
Зм.	Арк.	Прізвище	Підпис	Дата				
Розроб.		Оробей Е.В.			Автоматизована система управління технічним обслуговуванням виробничого обладнання	Літ .	Лист	Листів
Перевірів.		Новікова П.А.					2	9
Н. кон.		Телишева Т.О.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361		
Затв.		Павлов О.А.						

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Повне найменування системи та її умовне позначення

Повна назва системи: «Автоматизована система управління технічним обслуговуванням виробничого обладнання».

Коротке найменування системи: «service system».

1.2 Найменування організації-замовника та організації-учасника робіт

Замовником є кафедра автоматизованих систем обробки інформації і управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» (далі за текстом — Замовник).

Адреса замовника: м. Київ, п. Перемоги 37, 18 корпус ФІОТ.

Розробник сервісу — студент групи ІС-361 кафедри автоматизованих систем обробки інформації і управління Національного технічного університету України “Київський політехнічний інститут ім. Ігоря Сікорського” Оробей Едуард Віталійович.

1.3 Перелік документів, на підставі яких створюється система

При розробці системи і створення проектно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи.

Автоматизовані системи. Стадії створення;

- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплектність і позначення документів при створенні автоматизованих систем.

1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку роботи над створенням системи – 13 квітня 2020 року.

Плановий термін по закінченню роботи над системою – 2 червня 2020 року.

					ДП 6106.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

2 І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

2.1 Призначення системи

Автоматизована система управління технічним обслуговуванням виробничого обладнання призначена для підтримки діяльності ремонтно-експлуатаційних підрозділів виробництв.

2.2 Цілі та задачі створення системи

Цілями розробки автоматизованої системи управління технічним обслуговуванням виробничого обладнання є:

- Покращення контролю над технічним станом обладнання;
- Зменшення витрат часу на проведення технічного обслуговування та ремонту;
- Спрощення процедури замовлення запчастин та витратних матеріалів.

Головною ціллю створення інформаційної системи є підвищення ефективності використання обладнання на підприємстві за рахунок зменшення кількості виходів з ладу та зменшення часу простою обладнання в несправному стані.

Для досягнення цілей система має вирішувати наступні задачі:

- Ведення реєстрів обладнання;
- Ведення журналів технічного стану обладнання;
- Ведення реєстрів запчастин та витратних матеріалів;
- Планування робіт з технічного обслуговування;
- Автоматизація окремих етапів процедури замовлення запчастин.

3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Об'єктом автоматизації є процес організації та планування робіт ремонтно-експлуатаційного цеху

					ДП 6106.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

Процеси організації та планування робіт включають:

- Контроль за термінами технічного обслуговування;
- Створення списків обладнання, що потребує планового обслуговування;
- Редагування списків, відповідно до внесених даних;
- Генерація бланка замовлення запчастин;
- Сортування списків за оптимальністю виконання робіт;

Процеси аналізу отриманих даних включають у себе:

- Обчислення коефіцієнта оптимальності проведення робіт для кожного екземпляра обладнання.

					ДП 6106.01.000 ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

- Система має запускатися та коректно працювати на всіх сучасних операційних системах.
- Архітектура системи має бути гнучкою та придатною для розширення функціоналу.
- Система має надавати змогу зберігати, додавати, редагувати інформацію у реєстрах запчастин та обладнання.
- Система має відображати актуальну інформацію щодо стану системи.

4.2 Вимоги до надійності

Система повинна адекватно реагувати на помилки застосування та видавати відповідні повідомлення користувачеві.

4.3 Умови експлуатації

Для адекватної роботи системи необхідний пристрій з платформою, яка відповідає вимогам зазначеним в розділі 4.4 .

Усі користувачі системи повинні дотримуватися правил експлуатації електронної обчислювальної техніки.

4.4 Вимоги до складу і параметрів технічних засобів

Для правильної роботи даної програми до складу технічних засобів повинні входити:

Комп'ютер з конфігурацією не гірше:

- Процесор: Dual Core from Intel or AMD at 2.8 GHz;
- Об'єм оперативної пам'яті: 4 GB RAM;
- Відеокарта: Intel HD 3000;
- Storage: 450 MB available space.

Додатково має бути встановлене таке програмне забезпечення:

- Операційна система Windows XP або новіша;

					ДП 6106.01.000 ТЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

- DirectX: Version 9.0c.

Комп'ютерна периферія, до складу якої входить:

- Монітор;
- Клавіатура;
- Мишка.

					ДП 6106.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

У таблиці 5.1 наведено календарний план робіт та терміни їх виконання.

Таблиця 5.1 – Календарний план виконання робіт

№ з/п	Назва етапів створення продукту	Термін виконання етапу	Результат виконання
1.	Підготовка технічного завдання на розробку програмного продукту	13.04.2020	Технічне завдання підготовлено та узгоджено з замовником
2.	Розробка сценарію роботи	17.04.2020	Сценарій роботи розроблено
3.	Технічне проектування – функціональність, модулі, задачі, цілі тощо	23.04.2020	Цілі, задачі та функціональність визначені, технічне проектування завершено
4.	Узгодження з керівником інтерфейсу користувача	30.05.2020	Інтерфейс користувача узгоджено з керівником
5.	Алгоритмізація задачі	01.05.2020	Розроблено основний алгоритм роботи застосунку
6.	Розробка інформаційного забезпечення	05.05.2020	Розробка інформаційного забезпечення завершена
7.	Розробка програмного забезпечення	19.05.2020	Розробка програмного забезпечення завершена
8.	Налагодження програми	24.05.2020	Програму налагоджено
9.	Тестування програми	25.05.2020	Тестування завершено, застосунок успішно пройшов усі випробування
10.	Здача готового програмного продукту замовнику	30.05.2020	Готовий програмний продукт сдано замовнику

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ

6.1 Види випробувань

Види випробувань узгоджуються із замовником до проведення випробувань. Здача - прийом робіт виконується поетапно на комп'ютерах замовника в аудиторіях кафедри АСОІУ у відповідності з робочою програмою та календарним планом.

Усі програмні продукти, що створюються в рамках даної системи передаються замовнику як у вигляді готових модулів, так і у вигляді вихідних кодів, представлених в електронній формі.

					ДП 6106.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

Пояснювальна записка до дипломного проєкту

на тему: Автоматизована система управління технічним обслуговуванням
виробничого обладнання

Київ – 2020 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проєкту складається з 6 розділів, містить 21 малюнків, 13 таблиць, 1 додатків, 8 джерел.

Дипломний проєкт присвячений розробці автоматизованої системи управління технічним обслуговуванням виробничого обладнання. Цілями розробки є збільшення ефективності використання виробничого обладнання на підприємствах легкої промисловості за рахунок систематизації контролю над станом обладнання, планування проведення робіт з технічного обслуговування, контролю обігу витратних матеріалів та запчастин.

Задачі розробки:

Спроекувати базу даних обладнання та витратних матеріалів;

Розробити інтерфейс для обслуговуючого персоналу;

Розробити алгоритм планування проведення робіт.

У розділі інформаційного забезпечення описано вхідні дані, які отримує система, наведено структуру та опис бази даних системи, описано вихідні дані, які система надає користувачу.

Розділ математичного забезпечення присвячений розробці алгоритму планування проведення робіт з технічного обслуговування.

Програмне забезпечення . Систему розроблено на мові програмування C# з використанням .NET Framework та програмного забезпечення Visual Studio 2019 , Microsoft SQL Server 2019

У технологічному розділі наведено керівництво користувача

АВТОМАТИЗОВАНА СИСТЕМА, ЗАПЧАСТИНА, ОБЛАДНАННЯ, ЗАВДАННЯ, ТЕХНІЧНЕ ОБСЛУГОВУВАННЯ, ВИРОБНИЦТВО

					ДП 6106.00.000 ПЗ			
		<i>Прізвище</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>	Оробей Е.В.				Автоматизована система управління технічним обслуговуванням виробничого обладнання	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Перевірів.</i>	Новікова П.А.						2	57
<i>Н. кон.</i>	Телишева Т.О.					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361		
<i>Зав.</i>	Новікова П.А.							

ABSTRACT

Structure and scope of work. The explanatory note of the diploma project consists of 6 sections, contains 21 drawings, 13 tables, 1 applications, 8 sources.

The diploma project is devoted to the development of an automated control system for maintenance of production equipment. The objectives of the development are to increase the efficiency of the use of production equipment at light industry enterprises through the systematization of control over the condition of equipment, planning of maintenance work, control of the circulation of consumables and spare parts.

Development tasks:

Design a database of equipment and consumables;

Develop an interface for service personnel;

Develop an algorithm for planning work.

The information support section describes the input data received by the system, provides the structure and description of the system database, describes the output data that the system provides to the user.

The section of mathematical support is devoted to the development of an algorithm for planning maintenance work.

Software. The system is developed in the C # programming language using the .NET Framework and Visual Studio 2019 software, Microsoft SQL Server 2019

The technology section provides a user guide

KEY WORDS: AUTOMATED SYSTEM, SPARE PARTS, EQUIPMENT, TASKS,

MAINTENANCE, PRODUCTION

					ДП 6106.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

ЗМІСТ

ВСТУП	5
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	7
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА	8
1.1.1 <i>Опис процесу діяльності</i>	8
1.1.2 <i>Опис функціональної моделі</i>	12
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ	19
1.3 ПОСТАНОВКА ЗАДАЧІ	21
1.3.1 <i>Призначення розробки</i>	21
1.3.2 <i>Цілі та задачі розробки</i>	21
Висновок до розділу	22
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	23
2.1 ВХІДНІ ДАНІ	23
2.2 ВИХІДНІ ДАНІ	23
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ	23
Висновок до розділу	27
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	29
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ	29
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ	29
3.3 ОБГРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ	29
3.4 ОПИС МЕТОДУ РОЗВ'ЯЗАННЯ	30
Висновок до розділу	31
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	35
4.1 ЗАСОБИ РОЗРОБКИ	35
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	39
4.2.1 <i>Загальні вимоги</i>	39
4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	40
4.3.1 <i>Діаграма класів</i>	41
4.3.2 <i>Діаграма послідовності</i>	42
4.3.3 <i>Діаграма компонентів</i>	43
4.3.4 <i>Специфікація функцій</i>	44

					ДП 6106.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

Висновок до розділу	45
5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ	46
5.1 Керівництво користувача	46
5.2 Випробування програмного продукту	53
5.2.1 Мета випробувань.....	53
5.2.2 Загальні положення.....	53
5.2.3 Результати випробувань	54
Висновок до розділу	56
ЗАГАЛЬНІ ВИСНОВКИ	56
ПЕРЕЛІК ПОСИЛАНЬ	57
ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ.....	58

ВСТУП

Дипломний проєкт присвячений розробці автоматизованої системи управління технічним обслуговуванням виробничого обладнання. Розроблювана система призначена для використання на підприємствах легкої промисловості, а також будь-яких інших підприємствах розміром до 1000 працівників, де здійснюється робота з великою кількістю різнотипного обладнання та відбувається постійна зміна технологічних ліній та вимог.

З початку ХХ століття у світі спостерігається стрімке зростання чисельності міського населення, в тому числі за рахунок міграції, що є наслідком наукового та технологічного розвитку людства. Це супроводжувалось появою нових та зростанням розмірів існуючих міст. В останні десятиліття спостерігається стійка тенденція до зростання густоти населення, розширення території міст, утворення міських агломерацій. І саме в результаті збільшення густоти населення зменшується особистий простір кожного мешканця міста, зростає психологічний тиск. Як результат в кожного пересічного мешканця міста загострюється відчуття індивідуальності, бажання виокремитися з натовпу, що в свою чергу впливає на споживчі вподобання особи. Найбільш гостро це питання стоїть у галузі виробництва одягу та взуття, оскільки зовнішній вигляд особи є одним з найкращих інструментів для виокремлення з натовпу, демонстрації свого внутрішнього світу, можливістю певного самовираження. Як наслідок, стрімко змінюються тенденції та тренди на світових ринках відповідних груп товарів, що вимагає від виробників стрімко та динамічно реагувати на зміни.

Для задоволення потреб ринку одягу і взуття в умовах глобальної конкуренції, виробники зосереджують виробництво на невеликих фабриках, розміром до 1000 працівників, постійно розробляють нові моделі. Продукцію виготовляють невеликими партіями в залежності від поточної структури попиту на ринку. Це є досить ефективним підходом для задоволення потреб

					ДП 6106.00.000 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

сучасного споживача. Але недоліком такого підходу є потреба виробника у великій кількості різноманітного обладнання та постійній зміні виробничих ліній відповідно до структури замовлень. У виробництві використовується багато вузькоспеціалізованих видів обладнання, що мають різні експлуатаційні властивості. Окрім того, кожна окрема модель продукції вимагає певних специфічних налаштувань того чи іншого виду обладнання.

Кожен вид обладнання має певний перелік вимог, дотримання яких є необхідним для його ефективною експлуатації. Зокрема до таких вимог належать проведення періодичних технічних оглядів та обслуговування, відповідно до рекомендацій виробника обладнання. Завдання з дотримання експлуатаційних вимог обладнання на підприємстві покладається на окремий підрозділ – ремонтно-експлуатаційний цех. Вищеприписана модель виробництва передбачає періодичне використання певних видів обладнання. Тобто загальна кількість обладнання в 1.1 – 1.5 рази більша за кількість робітників, які на ньому працюють. В середньому на одного робітника ремонтно-експлуатаційного цеху припадає близько 50 – 90 одиниць обладнання, яке він має обслуговувати. В зв'язку з цим виникають великі труднощі з проведенням технічного обслуговування обладнання та підтриманням його в справному стані, динамічній зміні технологічних ліній відповідно до виробничого плану підприємства. Обладнання часто виходить з ладу внаслідок несвоєчасного проведення технічного обслуговування. Часто відсутні необхідні запчастини в потрібній кількості. Як наслідок, виробниче обладнання використовується не ефективно. Гарним підходом до вирішення вищевказаної проблеми є систематизація інформації про наявні типи, моделі, стан окремих екземплярів обладнання, обіг запчастин та витратних матеріалів, створення графіків планового технічного обслуговування обладнання.

Практичне значення одержаних результатів.

					ДП 6106.00.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

Для вирішення вищевказаної проблеми розроблено автоматизовану систему управління технічним обслуговуванням виробничого обладнання, яка систематизує інформацію про наявне на підприємстві виробниче обладнання, запчастини та витратні матеріали до нього, спрощує процедуру замовлення запчастин і витратних матеріалів. На основі отриманих даних створює графік проведення планового технічного обслуговування обладнання.

Визначення та терміни, що стосуються теми дипломного проєкту

Реєстр – упорядкована інформаційна система.

Технічне обслуговування – комплекс заходів з підтримки справності об’єкта при його експлуатації.

Список планових завдань – список завдань з технічного обслуговування, необхідність виконання яких була спрогнозована, виходячи з наявної інформації про обладнання.

Список поточних завдань – список завдань, що потребують виконання найближчим часом.

					ДП 6106.00.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Предметним середовищем розроблюваного проекту є можливість виробничого обладнання підприємства та працівників ремонтно – експлуатаційного цеху (далі – РЕЦ). Ремонтно – експлуатаційний цех – це підрозділ підприємства, що виконує роботи з технічного обслуговування та ремонту обладнання.

Однією з головних умов ефективної і стабільної роботи сучасних виробництв є підтримання в справному робочому стані виробничого обладнання. Для цього потрібне своєчасне проведення робіт з технічного обслуговування, з метою забезпечення дотримання характеристик параметрів обладнання в межах, вказаних виробником обладнання в технічному паспорті.

Автоматизована система управління технічним обслуговуванням виробничого обладнання має підвищувати ефективність роботи РЕЦ шляхом спрощення процедури замовлення запчастин і витратних матеріалів та систематизації інформації про наявне на підприємстві виробниче обладнання, запчастини та матеріали.

Основні завдання, що повинна виконувати інформаційна система:

- забезпечувати можливість планувати проведення робіт з технічного обслуговування обладнання;
- надати змогу користувачам переглядати інформацію про кількість, стан обладнання, наявність запчастин і витратних матеріалів до нього;
- налаштувати користувачам список завдань, які потребують виконання;
- спростувати процедуру замовлення запчастин.

При плануванні робіт з технічного обслуговування система повинна надавати перевагу виконанню завдань з вищим пріоритетом та складати

					ДП 6106.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

спикоск планових завдань таким чином, щоб протягом дня було виконано якнайбільше завдань.

1.1.1 Опис процесу діяльності

Автоматизації за допомогою розробленої інформаційної системи підлягає процес організації роботи ремонтно-експлуатаційного цеху підприємства та процес контролю за технічним станом і експлуатацією виробничого обладнання.

Підрозділ РЕЦ складається з механіків-налагоджувальників (далі – механік), що спеціалізуються на ремонті окремих видів систем (фахівці з ремонту пневматичних, гідравлічних, електричних систем) та типів обладнання (преси, холодильні установки, печі, транспортери, швейні машини та інше) і начальника РЕЦ.

Діяльність РЕЦ за відсутності інформаційної системи.

При відсутності завдань з технічного обслуговування механіки та начальник РЕЦ проводять візуальний огляд найважливішого обладнання. У разі виявлення несправностей проводиться пошук необхідних запчастин на складі механіком, виконуються ремонтні роботи. Якщо запчастина відсутня, обладнання виводиться з експлуатації, проводиться пошук обладнання для заміни в резерві. Якщо на початку дня є завдання з обслуговування, огляд обладнання не проводиться, доки всі завдання не будуть виконані. Огляд обладнання проводиться тільки за наявності вільного часу в механікв. Завдання на проведення робіт надходять від операторів обладнання та в результаті проведення оглядів. Начальник РЕЦ проводить перевірку виконання робіт. Перевіряє наявність запчастин на складі підприємства та складає бланк замовлення запчастин. Порядок виконання завдань визначається начальником РЕЦ.

На рисунках 1.1, 1.2 наведено діаграми активності за відсутності інформаційної системи та представлено у графічному додатку

					ДП 6106.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

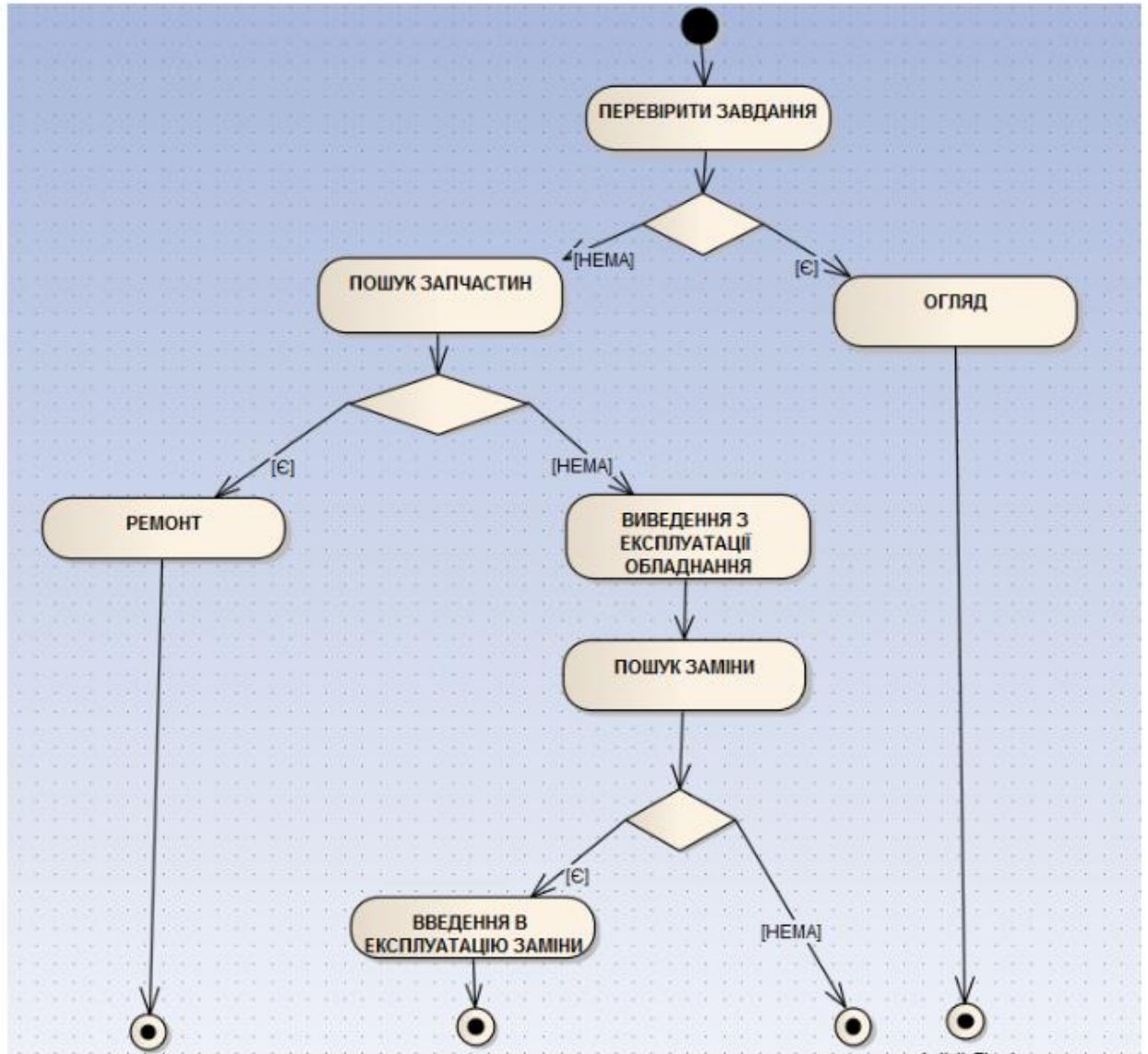


Рисунок 1.1 – Діаграма діяльності механіка за відсутності інформаційної системи.

Змн.	Арк.	№ докум.	Підпис	Дата

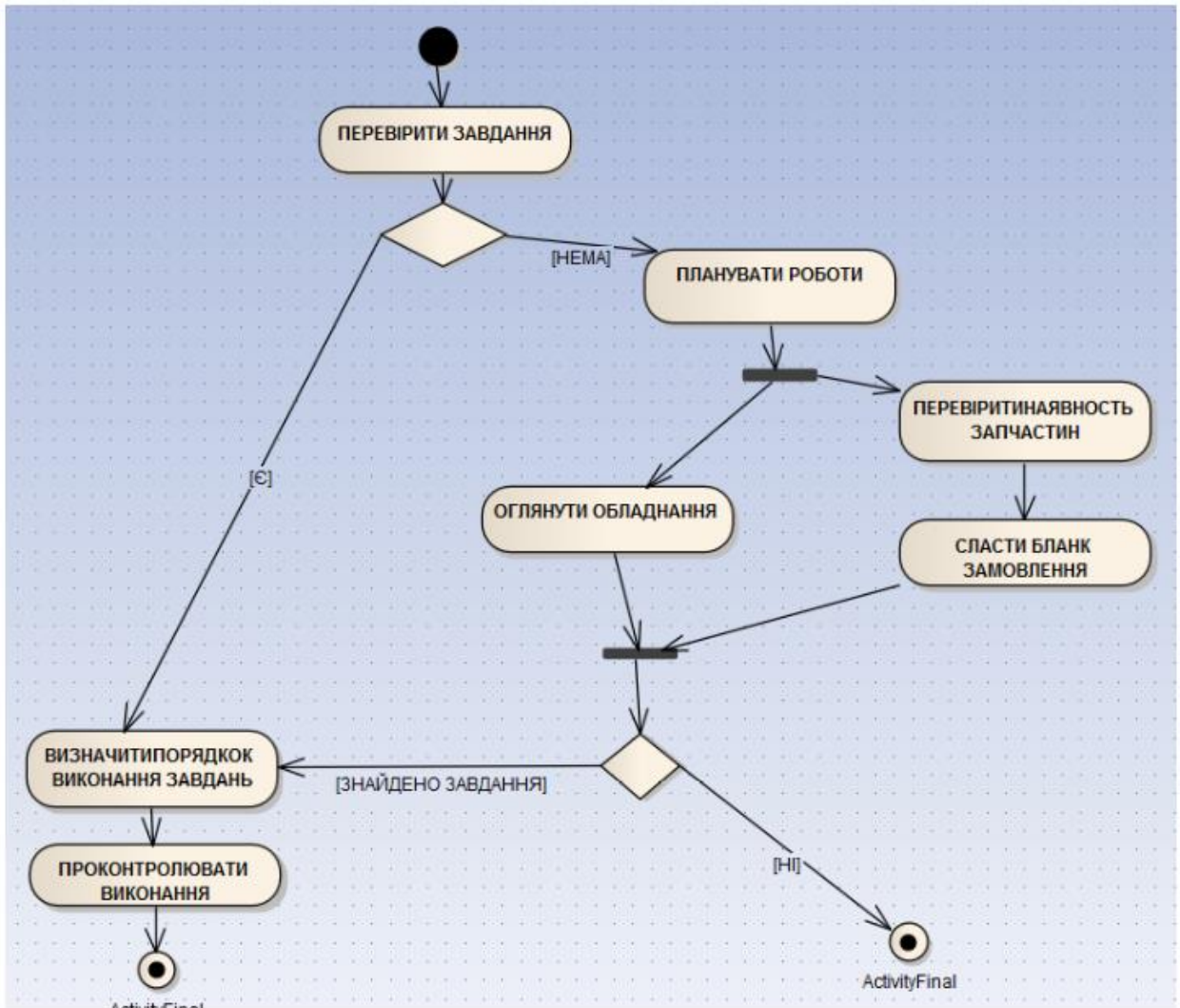


Рисунок 1.2 – Діаграма діяльності начальника РЕЦ за відсутності інформаційної системи.

Діяльність РЕЦ з використанням інформаційної системи.

Механік переглядає список завдань згенерованих системою. Якщо список не порожній, переглядає інформацію про наявність запчастин та аналогічного обладнання в резерві. Виконує завдання, після чого вносить дані про виконання, до системи. Після виконання кожного завдання механіком або начальником РЕЦ до системи вносяться дані про кількість використаних запчастин, матеріалів, зміни в складі обладнання. Завдання виконуються строго відповідно до порядку в списку. Також механік вказує в системі запчастини, які потрібно замовити.

Змн.	Арк.	№ докум.	Підпис	Дата

Начальник РЕЦ авторизується в системі, переглядає інформацію про наявні запчастини та коригує список запчастин, які потрібно замовити. Потім, після натиснення відповідної кнопки система генерує текстовий файл в якому міститься бланк замовлення. Детальний опис та перелік вхідних та вихідних даних наведено у розділі «Інформаціїе забезпечення».

За допомогою діаграм діяльності, наведених на рисунках 1.1 та 1.2 і представлених у графічному додатку, показано дії, які користувач повинен мати змогу виконати.

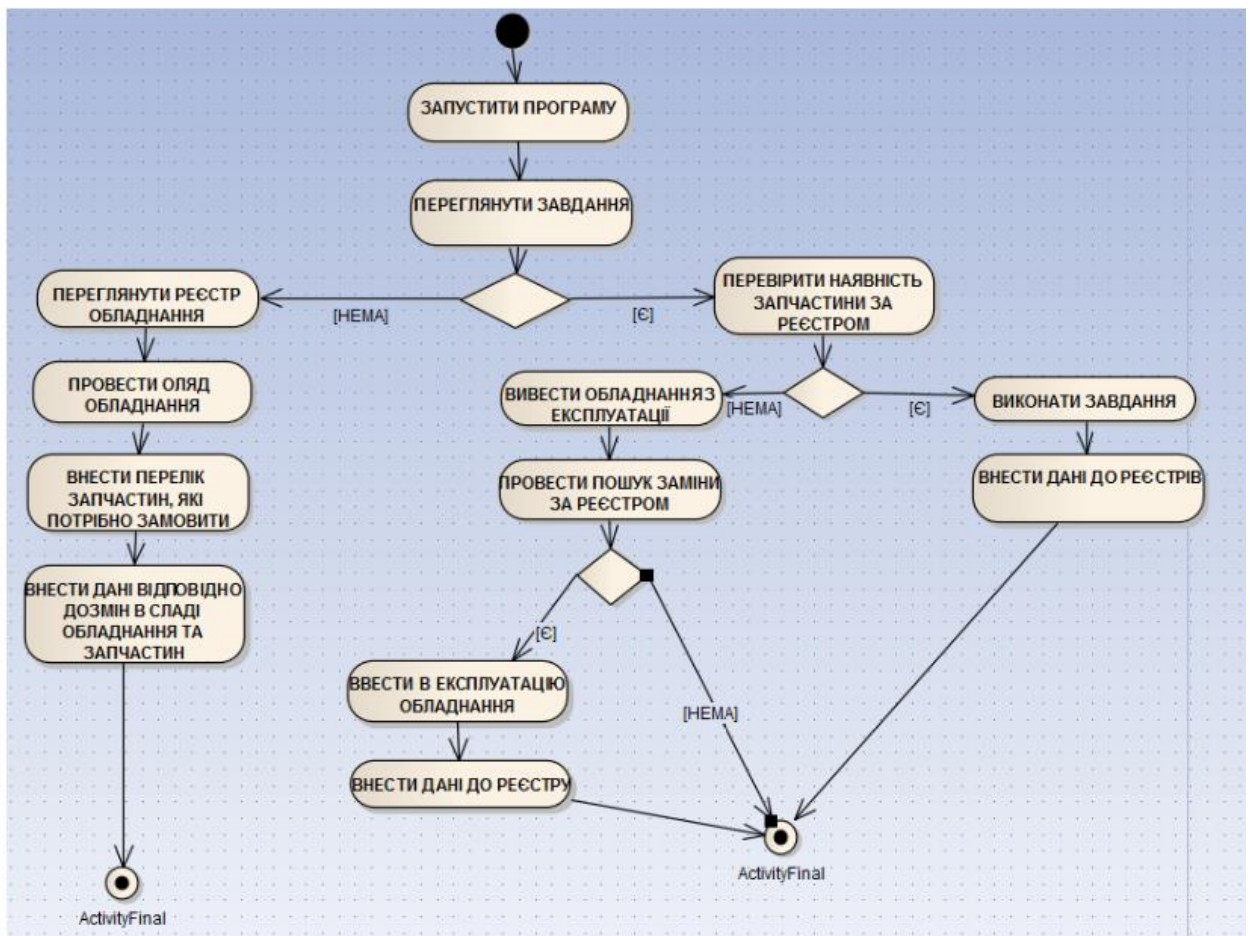


Рисунок 1.3 – Діаграма діяльності механіка

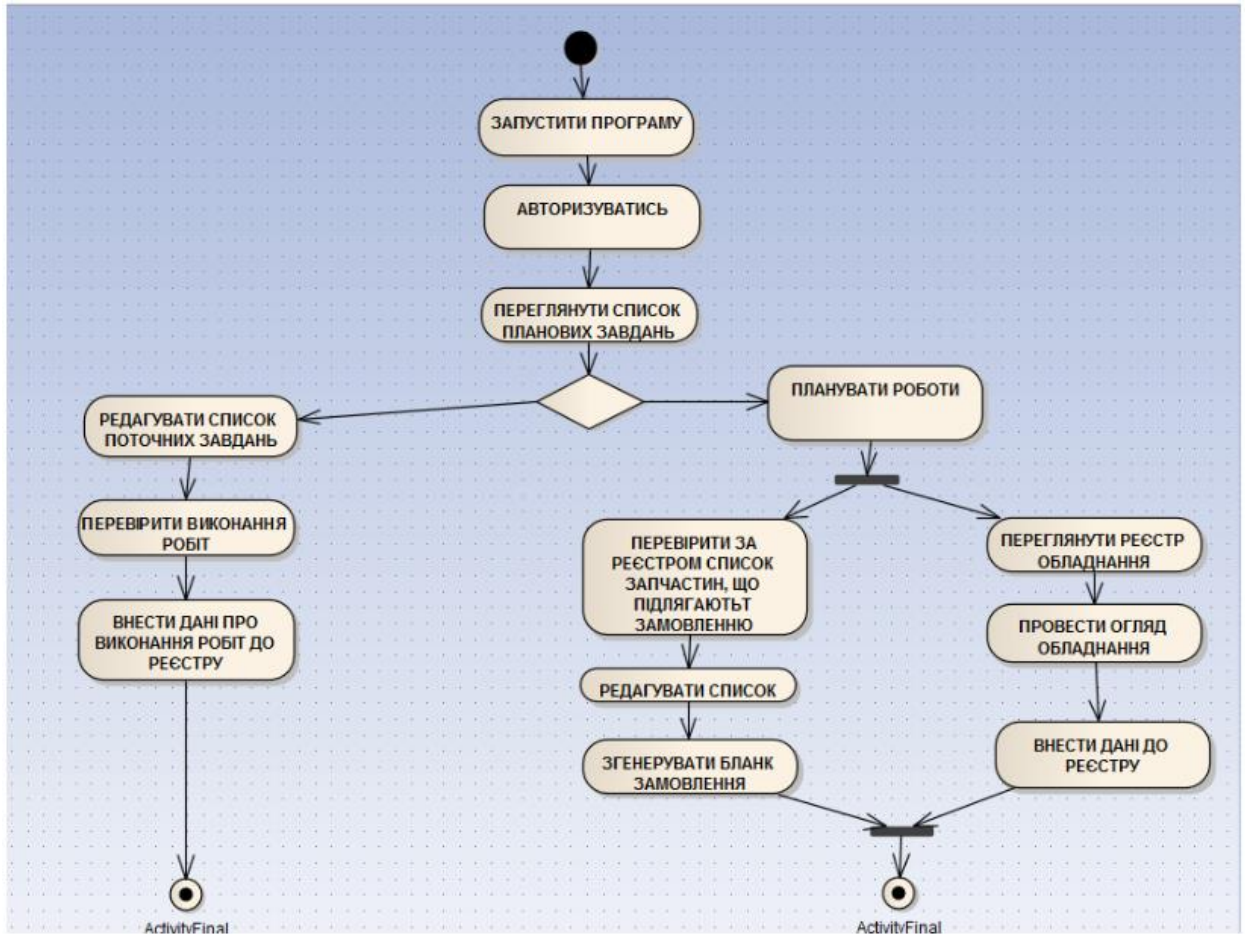


Рисунок 1.4 – Діаграма діяльності начальника РЕЦ

1.1.2 Опис функціональної моделі

Функціональна модель автоматизованої системи управління технічним обслуговуванням виробничого обладнання передбачає наявність двох типів акторів: механіки та начальник РЕЦ.

Механіки є співробітниками РЕЦ, які займаються введенням даних в систему, переглядом та пошуком даних, моніторингом (отриманням) списку завдань з планового технічного обслуговування.

Начальник РЕЦ – співробітник ремонтно-експлуатаційного цеху, який виконує адміністрування системи в цілому, моніторить та коригує список завдань, замовляє запчастини та витратні матеріали. Дані, які зберігає та обробляє система можна розділити на такі складові: реєстр обладнання, реєстр запчастин та витратних матеріалів, журнали технічного стану.

Варіанти використання доступні механіку:

- Перегляд списку поточних завдань;
- Внесення даних про виконані завдання;
- Перегляд реєстру запчастин;
- Складення переліку необхідних запчастин;
- Редагування реєстру запчастин;
- Перегляд реєстру обладнання;
- Редагування реєстру обладнання;
- Внесення даних про технічне обслуговування до журналів технічного стану обладнання;
- Перегляд журналів технічного стану обладнання;

Варіанти використання доступні начальнику РЕЦ:

- Перегляд списку планових завдань;
- Створення списку поточних завдань;
- Перегляд списку поточних завдань;
- Редагування списку поточних завдань;
- Перегляд реєстру обладнання;
- Редагування реєстру обладнання;
- Внесення даних про технічне обслуговування до журналів технічного стану обладнання;
- Перегляд реєстру запчастин;
- Редагування реєстру запчастин;
- Складення переліку необхідних запчастин;
- Замовлення запчастин;

Діаграма варіантів використання наведена на рисунку 1.3 та представлена у графічному додатку.

					ДП 6106.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

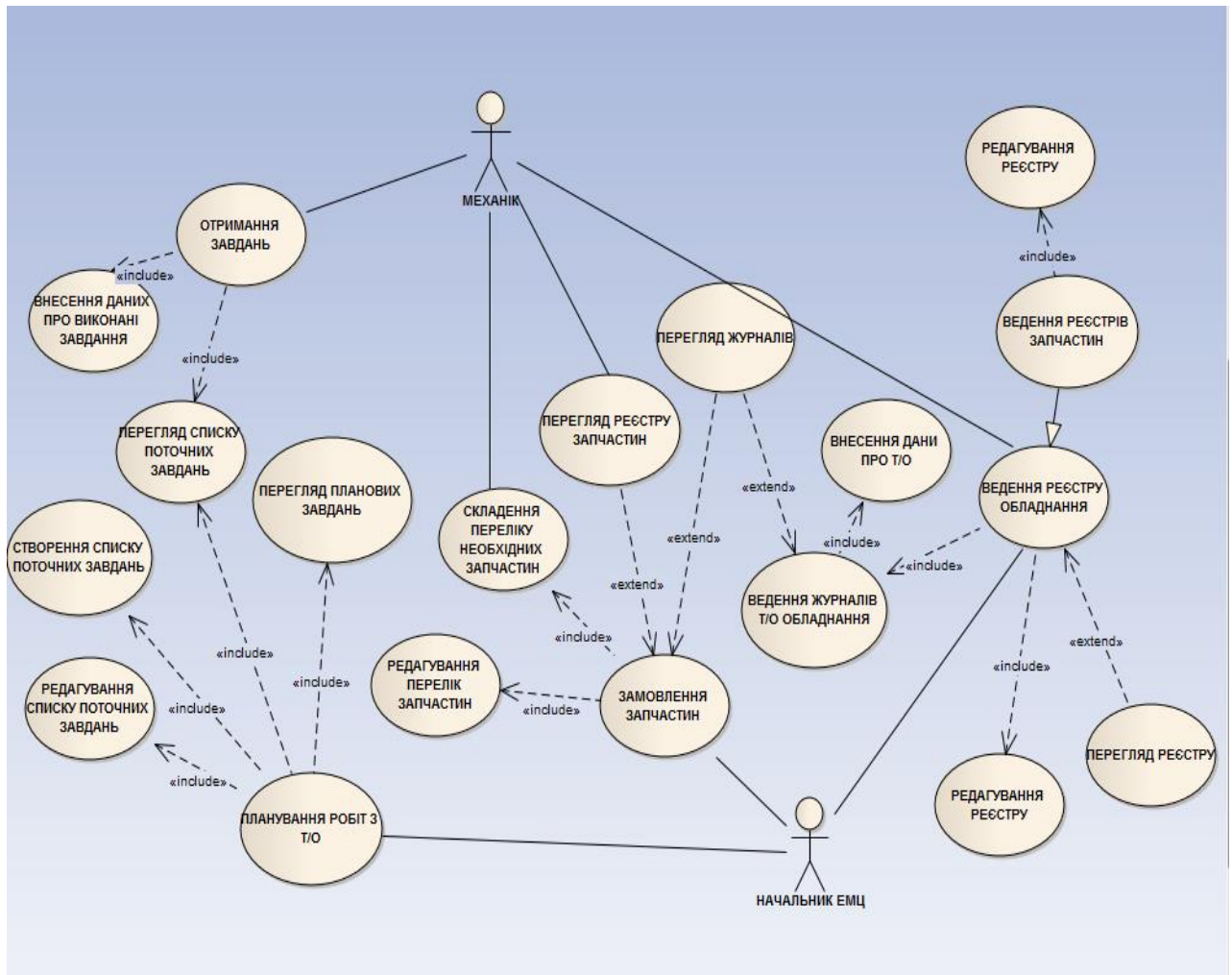


Рисунок 1.3 – Схема варіантів використання

Функціональні вимоги до варіантів використання наведено в таблиці 1.1

Таблиця 1.1 Вимоги до варіантів використання

Актор	Варіант використання	Опис вимог варіанта використання	Приоритет
Механік	Перегляд списку поточних завдань	RQ 001 Користувач отримує форму із списком поточних завдань RQ 001.1 В списку міститься інформація про тип обладнання, модель, серійний номер, відмітка про виконання	Високий Високий

Продовження таблиці 1.1

Актор	Варіант використання	Опис вимог варіанта використання	Приоритет
Механік	Перегляд списку поточних завдань	<p>RQ 001 Користувач отримує екранну форму що містить елемент в якому відображається список поточних завдань</p> <p>RQ 001.1 В списку відображається інформація про тип обладнання, модель, серійний номер, вид робіт, відмітка про виконання</p>	<p>Високий</p> <p>Високий</p>
Механік	Внесення даних про виконання завдання	<p>RQ 002 Користувач отримує екранну форму що містить елемент в якому відображається список поточних завдань</p> <p>RQ 002.1 Користувач ставить відмітку про виконання</p>	<p>Високий</p> <p>Високий</p>
Механік Начальник РЕЦ	Перегляд реєстру запчастин	RQ 003 Користувач отримує екранну форму що містить елемент в якому відображається реєстр запчастин	Високий

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 1.1

Актор	Варіант використання	Опис вимог варіанта використання	пріоритет
Механік	Складення переліку необхідних запчастин	RQ 004 Користувач отримує екранну форму що містить елемент в якому відображається реєстр запчастин RQ 004.1 користувач вносить відмітки до реєстру про необхідність замовлення	Високий Високий
Начальник РЕЦ, Механік	редагування реєстру обладнання	RQ 005 Користувач отримує екранну форму що містить елемент в якому відображається реєстр обладнання RQ 005.1 Користувач вносить актуальні дані до реєстру та зберігає їх	Високий Високий
Начальник РЕЦ, Механік	Перегляд реєстру обладнання	RQ 006 Користувач отримує екранну форму що містить елемент в якому відображається реєстр обладнання	Високий

Продовження таблиці 1.1

Актор	Варіант використання	Опис вимог варіанта використання	Приоритет
Начальник РЕЦ, Механік	Внесення даних до журналів технічного обслуговування обладнання	RQ 007 Користувач отримує екранну форму що містить елемент в якому відображається реєстр обладнання та поля для вибору журналу технічного обслуговування екземпляру обладнання	Високий
		RQ 007.1 Користувач заповнює поля (тип обладнання, модель, серійний номер)	Високий
		RQ 007.2 Користувач вносить записи до журналу	Високий
Механік	Редагування реєстру запчастин	RQ 008 Користувач отримує екранну форму що містить елемент в якому відображається реєстр запчастин	Високий
		RQ 008.1 користувач вносить актуальні дані до реєстру та зберігає їх	Високий
Начальник РЕЦ	Замовлення запчастин	RQ 009 Користувач отримує екранну форму з елементом в якому відображається реєстр запчастин та кнопка створення бланку замовлення у вигляді текстового файлу	Високий

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 1.1

Актор	Варіант використання	Опис вимог варіантів використання	Приоритет
Начальник РЕЦ	Редагування переліку необхідних запчастин	RQ 010 Користувач отримує екранну форму що містить елемент в якому відображається реєстр запчастин	Високий
		RQ 010.1 Користувач редагує відмітки про необхідність замовлення в реєстрі запчастин та зберігає дані	Високий
Начальник РЕЦ	Планування робіт з технічного обслуговування	RQ 011 Користувач отримує екранну форму що містить елемент в якому відображається список планових завдань згенерований системою та елемент, в якому відображається список поточних завдань	Високий
		RQ 011.1 Користувач створює або редагує список поточних завдань на основі списку планових та екстрених завдань	Високий

1.2 Огляд наявних аналогів

Розроблювана в даному проекті інформаційна система відноситься до класу CMMS (або КСУТО - комп'ютеризована система управління технічним обслуговуванням) – це комплекс програмного забезпечення, що включає базу

					ДП 6106.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

даних обладнання підприємства, модулі планування проведення робіт з технічного обслуговування та планово ремонту, оформлення заявок на проведення ремонту, модулі складського обліку, заявок на придбання запчастин і матеріалів, обліку людино-годин, ведення різноманітних статистик (витрати запчастин і матеріалів, витрати часу на виконання планових і позапланових ремонтів, витрати ресурсів на окремо взяті об'єкти, вузли, обладнання, техніку та інші цінності в ході технічного обслуговування). Розвинені системи CMMS, як правило, підтримують чітку ієрархічну систему користувачів, коли кожен з працівників компанії має доступ тільки до тих модулів, які безпосередньо необхідні в роботі. У межах кожного модуля користувач має права на дії відповідно до його профілю. Наприклад, виконавець нарядів на ремонт може бачити тільки завдання на проведення обслуговування того чи іншого обладнання; фахівець з обліку матеріальних цінностей може замовити необхідний інструмент, матеріали, запчастини, засоби індивідуального захисту; планувальник (оператор, диспетчер) - видати наряди на виконання ремонту, пріоритезувати завдання, змінити графік техобслуговування; співробітник відділу кадрів - додати або видалити співробітників в систему. Можливості CMMS визначаються спеціалізацією завдань. Розвинена логічна складова є однією з переваг системи. Система автоматично відпрацьовує заявки на запасні частини, витратні матеріали або обладнання, яке повинно бути замінено або використано згідно з графіками обслуговування, резервує їх на складі або видає сигнал на необхідність їх закупівлі. При цьому постійно ведеться підстроювання мінімально-максимального рівня наявності на складі обладнання та матеріалів, який би дозволив грамотно розподілити оборотні кошти, аби не заморозити їх на складі, але і не допустити простоїв через очікування поставки необхідних матеріалів. Ринок CMMS систем представлений:

ІС:ТОіР Управління ремонтами та обслуговуванням;

					ДП 6106.00.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

CalemEAM;
 Global-EAM;
 NERRA EAM;
 OpenMAINT;
 RealMaint TORO;
 Seascape;
 TRIM;

Історично CMMS системи розвивались великими корпораціями і були орієнтовані на використання на великих підприємствах, що налічували декілька тисяч співробітників.

Головними рисами проектованої системи є:

- простий дизайн, дружній до користувача;
- орієнтованість на використання в невеликих за розміром підприємствах головним чином легкої промисловості (розміром до 1000-1200 співробітників);
- орієнтованість на використання невеликими командами спеціалістів з технічного обслуговування;
- Відсутність потреби у потужних засобах технічного забезпечення;
- простота та гнучкість архітектури.

1.3 Постановка задачі

1.3.1 Призначення розробки

Автоматизована система управління технічним обслуговуванням виробничого обладнання призначена для підтримки діяльності ремонтно-експлуатаційного цеху виробництв.

1.3.2 Цілі та задачі розробки

Цілями розробки автоматизованої системи управління технічним обслуговуванням виробничого обладнання є:

					ДП 6106.00.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

- покращення контролю над технічним станом обладнання;
- зменшення витрат часу на проведення технічного обслуговування та ремонту;

- спрощення процедури замовлення запчастин та витратних матеріалів;

Головною ціллю створення інформаційної системи є підвищення ефективності використання обладнання на підприємстві за рахунок зменшення кількості виходів з ладу та зменшення часу простою обладнання в несправному стані.

Для досягнення поставлених цілей система має вирішувати наступні задачі:

- ведення реєстрів обладнання;
- ведення журналів технічного стану обладнання;
- ведення реєстрів запчастин та витратних матеріалів;
- планування робіт з технічного обслуговування;
- автоматизація окремих етапів процедури замовлення запчастин.

Висновок до розділу

В процесі написання розділу було проаналізовано предметну область, виявлено вже існуючі аналоги та сформульовано постановку задачі.

В розділі описано предметне середовище ремонтно-експлуатаційного цеху не лише в контексті проектування інформаційної системи, а й в цілому для розуміння необхідних для досягнення поставлених цілей задач, описана функціональна модель та приведено функціональні вимоги щодо роботи інформаційної системи.

Наведено діаграму варіантів використання, що показує функції, які може виконувати автоматизована система та розподіл функціональних складових між ролями співробітників ремонтно-експлуатаційного цеху.

В розділі приведені наявні аналоги та проведено аналіз над ними на наявність різних особливостей. Проаналізовано переваги даного продукту

					ДП 6106.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

над розробками конкурентів та обґрунтовано переваги даної системи над існуючими аналогами з технічної точки зору.

В розділі наведено призначення системи, описано цілі розробки та визначено необхідні задачі для коректної роботи системи відповідно до вимог функціональної моделі.

					ДП 6106.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідні дані до системи надходять від користувачів різних типів. Загалом обидва типа користувачів мають однакову можливість внесення до системи даних про стан обладнання, наявність запчастин, витратних матеріалів, обладнання. Дані про проведення екстрених (невідкладних) робіт включаються до списку поточних робіт виключно начальником РЕЦ. Відмітки про виконання робіт вносяться до системи виключно механіком. Детально вхідні дані описані в таблиці 2.1

Таблиця 2.1 Опис вхідних даних

Сутність яку описують дані	Джерело даних	Тип даних	перелік
обладнання	Механік – налагоджувальник , начальник РЕЦ	числовий	Кількість обладнання: справного; в резерві; даного типу; загальна; в ремонті; потребують відновлення; пріоритет ремонту. Періодичність проведення: Заміни тех. Рідин; оглядів; очищення радіаторів
		текстовий	Тип, модель, серійний номер, поточний статус (в ремонті, справний, в запасі)

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 2.1

Сутність яку описують дані	Джерело даних	Тип даних	перелік
	Дата/час	Дата виготовлення, останньої заміни технічних рідин, останнього огляду, останнього очищення радіаторів.	
запчастини	Механік – налагоджувальник, начальник РЕЦ	числовий	Кількість запчастин: Загальна; Рекомендована; Потрібно замовити. Орієнтовний термін служби.
		текстовий	Артикул запчастини, тип вузла обладнання, модель обладнання
Витратні матеріали	Механік – налагоджувальник, начальник РЕЦ	числовий	Середнє споживання, кількість наявна та яку потрібно замовити

Змн.	Арк.	№ докум.	Підпис	Дата

Кінець таблиці 2.1

Сутність яку описують дані	Джерело даних	Тип даних	перелік
		текстовий	Тип матеріалів, артикул
ремонт	Механік – налагоджувальник, начальник РЕЦ	числовий	Номер ремонту, код ремонту (вказує на тип проведених робіт), затрачений час,
		текстовий	Список деталей: замінені; які потрібно замінити. Серійний номер екземпляра обладнання
		Дата/час	Дата проведення ремонту
	Механік – налагоджувальник	Числовий (булевий)	Відмітка про виконання завдання
	начальник РЕЦ	Текстовий	Короткий опис завдання

Змн.	Арк.	№ докум.	Підпис	Дата

2.2 Вихідні дані

Вихідними даними системи є узагальнена інформація у реєстрах про поточний стан обладнання, кількість запчастин та витратних матеріалів (структуру та детальний опис реєстрів наведено в пункті 2.3), а також список планових завдань, поточних завдань(тип обладнання, модель, серійний номер, вид робіт), текстовий документ замовлення (бланк замовлення) запчастин та витратних матеріалів(артикул запчастини, кількість, модель обладнання (артикул). Опис структури списків завдань та текстового документа замовлення наведено в таблиці 2.2

Таблиця 2.2 Опис вихідних даних

Тип вихідних даних	Опис структури	Перелік полів	Опис полів	Як створюється
Узагальнена інформація в реєстрах	Відповідає структурі таблиць бази даних	Всі поля що містяться в базі даних	Наведено в розділі 2.3	Система відображає інформацію з бази даних
Список планових завдань	Має структуру таблиці	Тип обладнання; Модель обладнання; Серійний номер; Тип робіт; Короткий опис	Текстовий Текстовий Текстовий Числовий текстовий	Система створює список аналізуючи значення відповідних полів бази даних

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 2.2

Тип вихідних даних	Опис структури	Перелік полів	Опис полів	Як створюється
Сисок поточних завдань	Має структуру таблиці	Тип обладнання; Модель обладнання; Серійний номер; Тип робіт; Короткий опис	Текстовий Текстовий Текстовий Числовий текстовий	Список створюється начальником РЕЦ шляхом редагування списку планових завдань
Бланк замовлення	Містить дату складання та перелік запчастин у вигляді таблиці	Артикул запчастини; Кількість запчастин (однакових); Модель обладнання;	Текстовий Числовий Текстовий	Генерується системою за запитом користувача на основі наявної в базі даних інформації

2.3 Опис структури бази даних

В таблицях 2.3, 2.4, 2.5, 2.6, 2.7, 2.8 приведено опис структурних елементів бази даних.

Таблиця 2.3 загальна кількість обладнання відповідно до типу

Назва таблиці	Назва поля	Тип даних	Опис
TypeOfEquipment	Type_equipment	String	Тип обладнання
	total	Integer	Загальна кількість
	good	Integer	Кількість справних
	Under_repair	Integer	Кількість в ремонті

Таблиця 2.4 Дані обладнання відповідно до моделі

Назва таблиці	Назва поля	Тип даних	Опис
ModelOfEquipment	Model	String	Артикул моделі обладнання
	Type_equipment	String	Тип обладнання
	_total	Integer	Загальне число
	_good	Integer	справних
	Reserve	Integer	В запасі
	Bad	Integer	Потребують відновлення
	Liquid	String	Тип використовуваних технічних рідин

Таблиця 2.5 Дані по екземплярам обладнання

Назва таблиці	Назва поля	Тип даних	Опис
UnitsOfEquipment	Serial_number	String	Серійний номер екземпляра

Продовження таблиці 2.5

Назва таблиці	Назва поля	Тип даних	Опис
	produced	DateTime	Дата виготовлення
	Status	String	Поточний статус (справний, в ремонті, в резерві)
	priority	integer	Приоритет ремонту
	Repair_count	Integer	Лічильник ремонтів
	Fluid_replacement	Integer	Період роботи між заміною технічних рідин рекомендований виробником обладнання
	Date_f_r	Date	Дата останньої заміни тех. Рідин
	Time_clean	integer	Час потрібний на очистку радіаторів
	Time_liquid	integer	Час потрібний на заміну рідин
	Parts_control	Integer	Періодичність проведення планового огляду робочих органів обладнання (рекомендована)
	Date_p_c	Date	Дата останнього огляду
	Radiator_clean	Integer	Періодичність очищення радіаторів(рекомендована)
	Date_r_c	Date	Дата останнього очищення
	Model	String	Артикул моделі обладнання

Змн.	Арк.	№ докум.	Підпис	Дата

Таблиця 2.6 Дані по ремонтам обладнання

Назва таблиці	Назва поля	Тип даних	Опис
Repairs	N_repair	Integer	Номер ремонту
	Repair_code	Integer	Код ремонту
	Date_repair	Date	Дата проведення
	Time_repair	Integer	Затрачений час
	Replaced_parts	String	Замінені деталі
	Need_replacement	String	Деталі що потребують заміни
	Notes	String	Короткий опис завдання
	Serial_number	String	Серійний номер екземпляра

Таблиця 2.7 Реєстр запчастин

Назва таблиці	Назва поля	Тип даних	Опис
Spare_parts	Article	String	Артикул запчастини
	Total	Integer	Загальна кількість
	Type_of_units	String	Тип вузла обладнання
	Recommended_amount	Integer	Рекомендована кількість
	Service_life	Integer	Термін служби(орієнтовний)
	Model	String	Модель обладнання
	Need_to_order	Integer	Кількість запчастин ,яку потрібно замовити

Змн.	Арк.	№ докум.	Підпис	Дата

Таблиця 2.8 Дані про витратні матеріали

Назва таблиці	Назва поля	Тип даних	Опис
Supplies	Type	String	Тип витратних матеріалів
	Article	String	Артикул
	Total	Integer	Наявна кількість
	N_to_order	Integer	Кількість ,яку потрібно замовити
	Average_consumption	Double	Середнє споживання

Між таблицями бази даних реалізовано зв'язок один до багатьох, первинні ключі виділено жирним шрифтом.

Схема бази наведена на рисунку 2.1 та представлена у графічному додатку

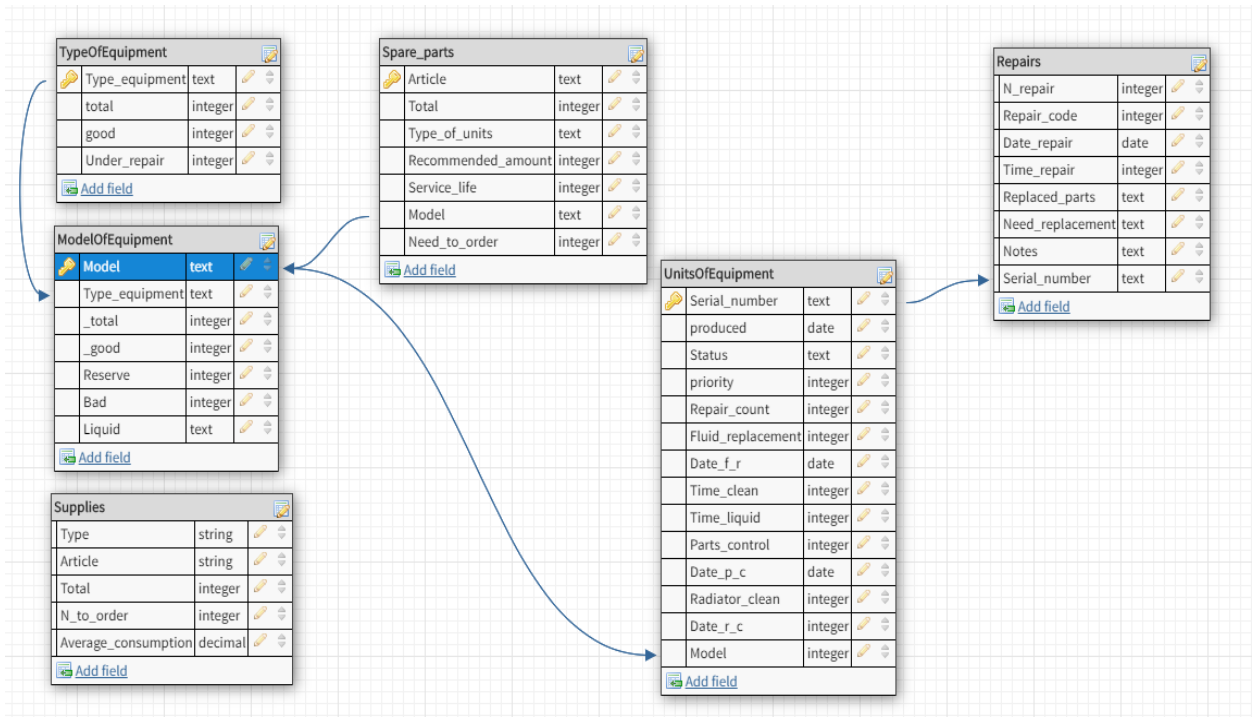


Рисунок 2.1 Схема бази даних

Висновок до розділу

В розділі наведено опис вхідних даних автоматизованої системи управління технічним обслуговуванням виробничого обладнання, наведено джерела вхідних даних. Описано вихідні дані, структуру вихідних даних, що являють собою таблиці.

У вхідних даних описані джерела вхідних даних, та яка саме інформація надходить з кожного з джерел.

Подано та описано перелік вихідних даних інформаційної системи, наведено їх структуру. Описано базу даних, представлено її структуру, структуру кожної з таблиць. Подано детальний опис змістової структури таблиць, наведено типи даних відповідних полів таблиці. Наведено змістовий (сутнісний) опис кожного поля з таблиць.

Приведено схему бази даних, що демонструє зв'язки між таблицями, зовнішні та внутрішні ключі, вказано тип зв'язків між таблицями (зв'язок один до багатьох).

Кількість та структура таблиць дозволяє інформаційній системі розв'язувати покладені на неї задачі та розширювати систему.

					ДП 6106.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Автоматизована система управління технічним обслуговуванням виробничого обладнання призначена для спрощення та покращення контролю над станом обладнання, та зменшення часу необхідного для проведення технічного обслуговування і як наслідок оптимізацію його використання. Таким чином, виходячи з архітектури системи, за допомогою математичних інструментів нам потрібно розв'язати задачу зменшення часу необхідного для проведення технічного обслуговування, тобто створити математичну модель та розробити алгоритм планування робіт з технічного обслуговування.

3.2 Математична постановка задачі

Задача полягає у тому, що необхідно знайти таке впорядкування списку завдань, при якому за певний проміжок часу буде обслуговано максимальну кількість екземплярів обладнання, відповідно до рівня пріоритету кожного екземпляра.

Вхідні дані:

- пріоритет (x);
- час на проведення певного типу робіт по кожному екземпляру (t);
- проміжок часу протягом якого проводяться роботи, тобто період роботи ремонтно-експлуатаційного цеху протягом дня (T).

3.3 Обґрунтування методу розв'язання

Вищевказана задача належить до класу задач багатокритеріального вибору.

Задача багатокритеріальної оптимізації формулюється таким чином:

					ДП 6106.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

$$\min_{\vec{x}} \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})\}, \quad (3.1)$$

$$\vec{x} \in S \text{ де } f_i : R^n \rightarrow R \text{ це } k (k \geq 2) \quad (3.2)$$

цільових функцій, вектори розв'язків

$$\vec{x} = (x_1, x_2, \dots, x_n)^T \quad (3.3)$$

належать до не порожньої області визначення S.

Задача багатокритеріальної оптимізації полягає у пошуку вектора цільових змінних, який задовольняє накладеним обмеженням та оптимізує векторну функцію, елементи якої відповідають цільовим функціям. Ці функції утворюють математичне описання критерію задовільності та, зазвичай, взаємно конфліктують. Звідси, «оптимізувати» означає знайти такий розв'язок, при якому значення цільових функцій були б прийнятними для постановника задачі [5].

Інколи застосовують методи скаляризації.

Оскільки цільова функція задачі багатокритеріальної оптимізації має векторні значення, її перетворюють на функцію зі скалярним значенням. Таким чином, задача багатокритеріальної оптимізації зводиться до задачі оптимізації з однією скалярною цільовою функцією. Одним із методів скаляризації є метод зваженої суми, який полягає в використанні наступної функції

$$F_1(\vec{f}(\vec{x})) = w_1 f_1(\vec{x}) + \dots + w_r f_r(\vec{x}). \quad [5] \quad (3.4)$$

В наведеній формулі (3.4) $f_i(x_i)$ – відповідні функції i - того критерію, w – вага критерію.

3.4 Опис методів розв'язання

В нашому випадку ми маємо два критерія, що можуть конфліктувати:

Пріоритет (x) та час (t) робіт, тому сформуємо цільову функцію, застосовуючи метод зваженої суми:

$$f(x,t) = x_1/t_1 + x_2/t_2 + \dots + x_n/t_n \quad [5], \quad (3.5)$$

яка підлягає максимізації, окрім того маємо очевидну умову:

$$T \geq t_1 + \dots + t_n \quad (3.6)$$

					ДП 6106.00.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Аналізуючи функцію $f(x,t)$ маємо наступні міркування: протягом періоду T може бути виконано не більше ніж

$$y = T \text{ div } \min(t) \quad (3.7)$$

тобто цілочисельного ділення T на мінімальне з усіх t .

Нехай:

$$S(t) = \sum^i t_i \quad (3.8)$$

Таким чином якщо сума всіх t $S(t) > T$ то список завдань не може бути виконаний повністю, якщо $S(t) \leq T$ то список завдань буде виконаний повністю незалежно від впорядкування. Таким чином задача зводиться до обчислення критерію:

$$k = x/t \quad (3.9)$$

та сортування списку завдань (наприклад методом сортування вибором, оскільки в нас лінійний масив невеликого розміру і швидкість сортування нам не принципова) за критерієм k в порядку спадання, тобто чим вищий критерій k тим більш пріоритетним є завдання.

Висновок до розділу

В цьому розділі було сформульовано змістовну та математичну постановку задачі, визначено клас до якого відноситься задача. Також було визначено базисні змінні (тобто дані, які вичерпно визначають значення параметра, що підлягає оптимізації).

Було розглянуто частину методів, що розроблені для розв'язання визначеного класу задач, сформульовано математичну модель задачі, проведено її аналіз, в результаті якого знайдено ефективний та простий алгоритм розв'язання задачі, який досить просто реалізувати програмними методами.

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Інформаційну систему було створено на мові програмування С# з використанням таких засобів розробки як Microsoft Visual Studio 2019 з розширенням Microsoft.NET Framework та база даних Microsoft SQL Server 2019.

С# проста та повноцінна високорівнева мова програмування, що пропонує значну кількість інструментів для структурування та підтримки великих програм і рішень. В ній краще, порівняно з С/С++ реалізовано обробку помилок, вона дозволяє використовувати типи даних високого рівня, такі як гнучкі масиви, списки і словники, для ефективної реалізації яких на С/С++ потрібні суттєві затрати часу. З метою розширення функціональності зручно використовувати готові бібліотеки, заватаження яких зручно організовано шляхом імпортування безпосередньо в середовище розробки через вбудований у Visual Studio 2019 менеджер пакетів NuGet Package Manager .

С# дає можливість розділяти програми на модулі, які пізніше можуть бути застосовані в інших програмах, чим підтримується багаторазовість застосування коду. Під час розробки великих рішень зазвичай програма розділяється на окремі проекти, котрі відповідають за певний клас задач подібної направленості, такий підхід використовується з метою спрощення орієнтування в рішенні та структурування рішень. Разом з С# надається значна кількість стандартних бібліотек, які зручно використовувати і як основу для розробки нових застосунків і як приклади при вивченні мови. Стандартні модулі містять засоби для роботи з системними викликами, файлами, мережними з'єднаннями та інтерфейсами до різних графічних бібліотек.

					ДП 6106.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

C# є інтерпретованою мовою і завдяки цьому вдається заощадити час, який зазвичай втрачається при компоновці та компіляції рішень. Є можливість інтерактивного використання інтерпретатора, це уможливило проведення експериментів з можливостями мови C#, завдяки чому можливо писати зручні для читання програми, що дозволяє використання загальноприйнятих узгоджень стосовно написання коду та назв полів різних типів. Тексти програм написаних на високорівневих мовах, таких, як C# є коротшими ніж їхні еквіваленти на C/C++ з наступних причин:

- типи даних високого рівня абстракції дозволяють виражати складні операції однією інструкцією;
- великий вибір методів та структур;
- наявність сучасніших методів.

C# (як і Java) є представницею великої родини мов програмування що мають подібний до C++ синтаксис. Для мови характерні строга статична типізація, наслідування, перевантаження операторів, підтримка поліморфізму, інкапсуляції, закриття методів, вказівників на функції та члени класів, атрибутів, подій, властивостей, делегатів, винятків, коментарів у форматі XML. При розробці C# перейняла багато від своїх попередників — мов Java, C++, Delphi і Smalltalk і, виходячи з досвіду їхнього використання, виключає деякі конструкції, які виявились проблематичними при розробці програмних продуктів, як приклад множинне спадкування класів (на відміну від C++).

Програма написана з використанням Microsoft.NET Framework, на будь-якій підтримуваній фреймворком мові програмування, спочатку перекладається компілятором в єдиний для Microsoft.NET проміжний байт-код Common Intermediate Language (CIL) (раніше називався Microsoft Intermediate Language, MSIL). У термінах Microsoft.NET виходить збірка, англ. assembly. Потім код або виконується віртуальною машиною Common Language Runtime (CLR), або транслюється утилітою NGen.exe в виконуваний код для конкретного цільового процесора.

Використання віртуальної машини переважно, оскільки позбавляє розробників від необхідності піклуватися про особливості апаратної частини. У разі використання віртуальної машини CLR вбудований в неї JIT-компілятор «на льоту» (just in time) перетворює проміжний байт-код в машинні коди потрібного процесора. Сучасна технологія динамічної компіляції дозволяє досягти високого рівня швидкодії. Віртуальна машина CLR також сама піклується про базову безпеку, управлінні пам'яттю і системі винятків, позбавляючи розробника від частини роботи [2].

У специфікації Common Language Infrastructure (CLI), розробленій Microsoft і затвердженій ISO і ECMA, описана архітектура Microsoft.NET Framework, описано формат метаданих про структуру програми, система виконання байт-коду і багато іншого.

В бібліотеці Framework Class Library (FCL) містяться об'єктні класи, доступні для всіх підтримуваних мов програмування. До складу FCL входять класи Windows Presentation Foundation, ASP.NET, Language Integrated Query, ADO.NET, Windows Forms, Windows Communication Foundation та інші. Ядро FCL називається Base Class Library (BCL)

MS SQL Server — компактний багатопотоковий сервер баз даних. До головних переваг якого можна віднести велику швидкість, стійкість та простоту використання і найбільшою інтеграцією з іншими продуктами Microsoft .

MS SQL Server було розроблено компанією Microsoft з метою покращення швидкодії обробки великих баз даних і швидкого розгортання великих масивів даних.

MS SQL Server вважається вдалим рішенням для великих і середніх програмних продуктів. Вихідні коди сервера компілюються на багатьох платформах. Найкраще можливості сервера розкриваються в UNIX-системах, які мають підтримку багатопотоковості, завдяки чому зростає загальна продуктивність системи.

MS SQL Server є безкоштовним у використанні, окрім Professional та Enterprise версій продукту. До можливостей сервера MS SQL Server слід віднести:

наявність простої і ефективної системи безпеки;

простота у встановленні та використанні;

висока швидкість виконання команд;

кількість користувачів, які одночасно працюють із базою даних може налаштовуватися та обмежується виключно ресурсами сервера на якому розгорнута база даних;

кількість рядків у таблицях може досягати 50 млн.;

ADO.NET (ActiveX Data Object для .NET) - технологія, що надає доступ і керування даними, що зберігаються в базі даних або інших джерелах (Microsoft SQL Server, Microsoft Access, Microsoft Excel, Microsoft Outlook, Microsoft Exchange, Oracle, OLE DB, ODBC, XML, текстові файли), заснованих на платформі Microsoft.NET Framework і входить до складу .NET Framework 2.0, являє собою набір бібліотек [3].

Відмінністю ADO.NET від технології ADO, яка була орієнтована на використання при створенні тісно пов'язаних клієнт-серверних систем, є націленість на автономну роботу за допомогою об'єктів DataSet. Об'єкти DataSet представляють собою локальні копії взаємопов'язаних таблиць бази даних. Об'єкти DataSet надають можливість збірці, яка їх використовує (на зразок програми, виконуваної на комп'ютері) отримувати доступ до свого вмісту та змінювати його, не потребуючи підключення до джерела даних, та відправляти назад блоки змінених даних для обробки за допомогою відповідного адаптера. Найголовніша відмінність ADO.NET від ADO полягає в тому, що ADO.NET є керованою кодовою бібліотекою і керується тими ж правилами, що і будь-яка керована бібліотека. Типи, які є складовими ADO.NET, застосовують протокол управління пам'яттю CLR, належать до тієї ж системи типів (інтерфейси, делегати, класи, перерахування та структури).

Доступ до них можливий з використанням будь-якої з мов, підтримуваних .NET. Класи ADO.NET містяться у збірці SystemData.dll.

Entity Framework – бібліотека, яка забезпечує просте підключення до бази даних та дозволяє дублювати сутності з застосування . Ця бібліотека має декілька підходів, а саме:

Database first – підхід, при використанні якого, спочатку створюється база даних з усією її структурою. Потім на основі неї у кодї програми генерується набір сутностей, що повторюють структуру таблиць у базі даних та зв'язки між ними, створюючи при цьому навігаційні поля сутностей. Завдяки цьому з'являється можливість швидко отримувати доступ з однієї сутності до іншої, що пов'язана з нею за допомогою зовнішнього ключа у базі даних. Якщо змінюється структура бази даних під час розробки, то є можливість оновлення набору сутностей і зміни будуть перенесені в код програми.

Code first – підхід, при якому спочатку створюються сутності у кодї програми. Потім виходячи з їхньою структури, генерується та наповнюється структура бази даних зі зв'язками, зовнішніми та первинними ключами. Якщо під час розробки, у кодї відбувається зміна моделі, то існує можливість робити, так звані міграції на базу, котрі переносять зміни з моделі на базу без втрат даних, якщо відсутній конфлікт з існуючими даними;

4.2 Вимоги до технічного забезпечення

4.2.1 Загальні вимоги

Для повноцінної роботи інформаційної системи до складу технічних засобів повинні входити:

Комп'ютер користувача:

комп'ютер з такою конфігурацією:

процесор з тактовою частотою не нижче 2.6 ГГц;

об'єм вільного місця на жорсткому диску не менше 450 МБ;

					ДП 6106.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

достатній об'єм оперативної пам'яті (не менше 2ГБ);
 вимоги до параметрів інших складових не задаються, оскільки вони не значним чином впливають на роботу програми;
 має бути встановлене таке програмне забезпечення:
 операційна система Windows XP/Vista/7/8/8.1/10;
 комп'ютерна периферія, до складу якої входить:
 монітор;
 мишка;
 клавіатура.

4.3 Архітектура програмного забезпечення

При створенні інформаційної системи було використано наступний архітектурний підхід: все застосування розділено на три шари.

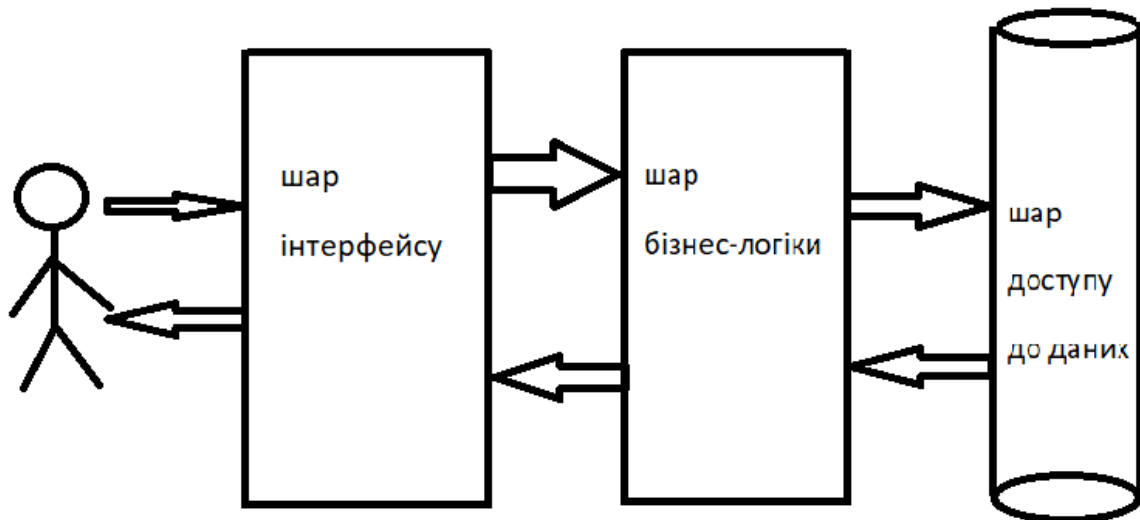


Рисунок 4.1 – Схема архітектури системи

Розглянемо кожен з них:

- шар інтерфейсу (UI), що дозволяє користувачу ефективно взаємодіяти з інформаційною системою. Цей шар дозволяє переглядати вихідні дані

системи та передає внесені користувачем дані до інших шарів, зокрема до шару доступу до даних, на яких програма розпізнає необхідні поля і перевіряє їх на коректність та, у випадку невалідності даних, повертає користувачеві повідомлення про помилку. Клієнтський шар побудований на основі технології WindowsForms. На даному рівні знаходиться перший з рівнів захисту інформаційної системи, а саме захист від несанкціонованого доступу до даних програми за відсутності відповідних прав користувача;

- шар бізнес логіки (Business logic) реалізує бізнес логіку інформаційної системи та функціонал системи. Даний рівень включає в собі всі функції та перетворення, що їх може виконувати система і виконує роль зв'язного компонента між рештою шарів системи. Реалізацію виконано за допомогою бібліотеки класів;

- шар доступу до даних (Data access) реалізує доступ до бази даних. Даний шар взаємодіє з базою даних за допомогою бібліотеки Entity Framework, включає в себе інформацію про всі таблиці у сховищі, у вигляді дублікатів класів, покращує взаємодію з базою даних та підвищує надійність збереження даних.

В разі виникнення помилок на будь-якому рівні вони ловляться та обробляються шаром бізнес логіки та показують користувачеві, що сталося в системі.

База даних інформаційної системи була спроектована таким чином, щоб мати змогу працювати з великими масивами даних без помітних втрат у швидкодії, що було досягнуто створенням таблиць не тільки за призначенням, а й за властивостями.

Автоматизована система управління технічним обслуговуванням виробничого обладнання за своєю суттю може розглядатися як підсистема, що входить до складу системи управління виробництвом та бути використана як складова ERP систем.

					ДП 6106.00.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

4.3.2 Діаграма послідовності

На схемі структурній послідовності розглянуто процеси перегляду реєстру та замовлення запчастин за допомогою класів, котрі зображено на рисунку 4.1.

Таблиця 4.1 – Об’єкти діаграми послідовності

Клас	За що відповідає
Form1	Головна форма.
Form2	Форма, що дозволяє переглядати реєстр обладнання
Form3	Форма, що дозволяє переглядати реєстр запчастин
Orderparts	Клас відповідальний за створення документів.
UnitOfEquipment	Клас, який відповідає сутності екземпляр обладнання

Структурна схема послідовності процесів рівня бізнес логіки зображена на рисунку 4.2 та представлена у графічному додатку

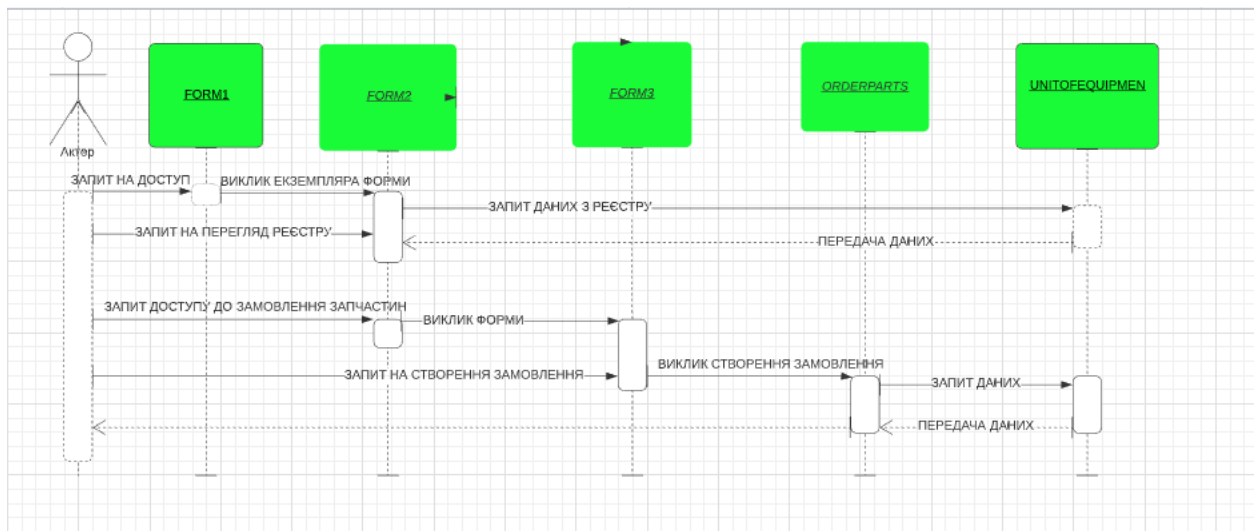


Рисунок 4.2 - Схема послідовності процесів бізнес логіки

4.3.3 Діаграма компонентів

4.3.4 Діаграма компонентів показує розбиття програмної системи на структурні складові та залежності між компонентами. На діаграмі зображено об'єкти, які представлені в таблиці 4.2.

4.3.5 Таблиця 4.2 – Об'єкти діаграми компонентів

Компонент	Опис компонента
UI	Відображення інтерфейсу користувача
DB	Збір та збереження даних
TaskPlaning	Планування робіт
Analysis	Аналіз даних

Діаграма компонентів зображена на структурній схемі компонентів на рисунку 4.3.

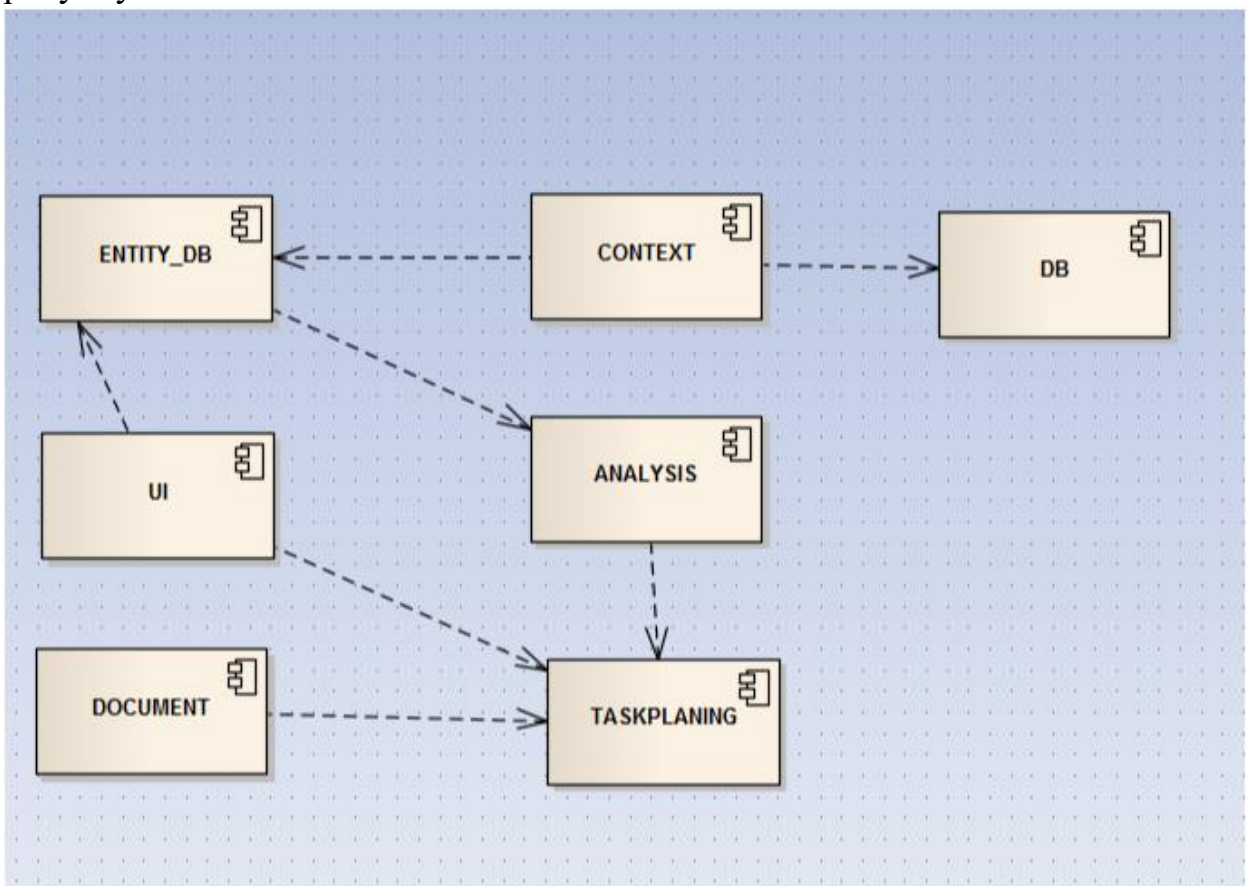


Рисунок 4.3 - діаграма компонентів

4.3.6 Специфікація функцій

Функції класів програмного забезпечення наведені в таблиці 4.3.

Таблиця 4.3 – Функції класів програмного забезпечення

Назва	Опис
public void Commit()	Зберігає зміни до бази даних.
public bool IsValidLoginData(Account account)	Перевіряє чи правильні дані профілю користувача.
public void UpdatePassword(string username, string password)	Змінює пароль.
public void SendParts(List<Part> Parts)	Створює бланк замовлення
public List<Task> GetTask ()	Дістає інформацію про планові роботи
Public List<Parts>GetParts()	Шукає запчастини
Public List<Units>GetUnits()	Шукає обладнання
public void AddParts(Parts parts)	Додає запчастини
public void AddUnits(Units units)	Додає обладнання
public int Sort(int k, int m)	Сортує список робіт

Висновок до розділу

У розділі ретельно описано засоби розробки які використані під час процесу створення автоматизованої системи управління технічним обслуговуванням виробничого обладнання та наведено головні переваги кожного з засобів. Було зроблено детальний опис кожного із програмних засобів, котрі застосовувались при розробці інформаційної системи та наведений опис функціонування її компонентів та які підходи потрібно використовувати при роботі з ними.

Наведено перелік вимог до технічного забезпечення, які необхідні для використання інформаційної системи.

В розділі описано архітектуру інформаційної системи та наведено обґрунтування вибору такого архітектурного підходу по кожному з рівнів. Обґрунтовано оптимальність обраної архітектури для побудови інформаційної системи, наведено переваги які отримує система при використанні обраного типу архітектури.

в розділі приведено структурні схеми класів, схеми послідовності та компонентів, котрі описують архітектуру системи і її частин. Описано частину функцій рівня бізнес логіки системи для більш повного розуміння виконуваних системою задач.

Вимоги, що прописані у технічному забезпеченні, накладають обмеження на використання продукту. Якщо середовище не відповідає вимогам, система не зможе функціонувати.

					ДП 6106.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Щоб почати роботу з системою потрібно на робочому столі комп'ютера, на який встановлена система обрати відпорний ярлик і двічі натиснути на нього лівою кнопкою маніпулятора миша.

Після цього на екрані відобразиться головна екранна форма інформаційної системи, на якій міститься меню. Вигляд головної форми показано на рисунку 5.1 та наведено в графічних матеріалах



Рисунок 5.1 – головна форма

Далі потрібно обрати подальший варіант використання шляхом натискання на кнопку з відповідним написом: «ОБЛАДНАННЯ»

«ЗАВДАННЯ», «ВИТРАТНІ МАТЕРІАЛИ» та «ВХІД ШЕФ РЕЦ»

Для перегляду та редагування реєстру обладнання або запчастин потрібно натиснути кнопку «ОБЛАДНАННЯ», в результаті виведеться форма для перегляду реєстрів

Змн.	Арк.	№ докум.	Підпис	Дата

Вигляд екранних форм показано на рисунках 5.2 – 5.11 та наведено в графічних матеріалах

	КодТовара	Марка	СерийныйНомер	НаСкладе	Цена	Заказано	МинимальныйЗап	ВремяЗадержки	КодТипа
*									

Рисунок 5.2- форма для перегляду та редагування реєстрів

Далі для перегляду реєстру потрібно натиснути кнопку з назвою відповідного реєстру, після чого буде виведено реєстр в нижній частині форми

	КодТовара	Марка	СерийныйНомер	НаСкладе	Цена	Заказано	МинимальныйЗап	ВремяЗадержки	КодТипа
▶	1	strobел	15253897	3	25				
	2	strobел	451200158	2	256000	0	1	5мес	5
	3	zic-zac	258035480	1	150000	1	1	12мес	3
	6	fortuna	2522364	5	522	0	2	2	8
	7	гншгн	46848	5					
	8	гцьшгрм	54646	52					
	9	ппгн	9	45					
	10	ро	78778	6					
	11	вьжь	л	6467					
	12	uybiu							
	13			45					
*									

Рисунок 5.3 – вивід реєстра

Для редагування реєстру дані вносяться до відповідної комірки після чого потрібно натиснути кнопку «ЗБЕРЕГТИ» для збереження даних.

Для додавання запчастин або обладнання до реєстру необхідно натиснути кнопку з відповідним написом, після чого буде виведена відповідна екранна форма



ТИП ОБЛАДНАННЯ

МОДЕЛЬ

АРТИКУЛ

КІЛЬКІСТЬ

ЗБЕРЕГТИ ВИХІД

Рисунок 5.4 – форма для додавання запчастин

ТИП ОБЛАДНАННЯ

МОДЕЛЬ

СЕРІЙНИЙ НОМЕР

ДАТА ВИГОТОВЛЕННЯ

СТАН

ВИХІД **ПЕРЕГЛЯНУТИ РЕЄСТР** **ЗБЕРЕГТИ**

Рисунок 5.5 – форма для додавання обладнання

Відповідно для додавання до реєстру нових типів запчастин або обладнання потрібно заповнити відповідні поля на формі та натиснути кнопку «ЗБЕРЕГТИ».

Для перегляду списку поточних завдань або пошуку потрібних запчастин або обладнання, потрібно натиснути кнопку «ЗАВДАННЯ», після чого буде виведена відповідна форма (мал 5.6)



Рисунок 5.6 – форма для перегляду завдань та пошуку запчастин або обладнання в реєстрі

При натисненні на кнопку «ПЕРЕГЛЯНУТИ СПИСОК ПОТОЧНИХ ЗАВДАНЬ» в нижній частині форми буде виведено відповідний список



Рисунок 5.7- перегляд списку поточних завдань

Для пошуку потрібних запчастин або обладнання треба натиснути на кнопку з відповідним написом, після чого буде виведено форму для пошуку (мал. 5.8)

Рисунок 5.8 – форма для пошуку запчастин або обладнання

Для перегляду або пошуку витратних матеріалів потрібно натиснути кнопку «ВИТРАТНІ МАТЕРІАЛИ», буде виведено форму (рис. 5.9)

	КодТовара	Марка	СерийныйНомер	НаСкладі	Цена	Заказано	МинимальныйЗап	ВремяЗад
▶	6	fortuna	2522364	5	522	0	2	2
	1	strobel	15253897	3	25			
	2	strobel	451200158	2	256000	0	1	5мес
	3	zic-zac	258035480	1	150000	1	1	12мес
*								

Рисунок 5.9 – форма для перегляду реєстру та пошуку витратних матеріалів

Для використання системи користувачем з правами начальника РЕЦ спочатку потрібно авторизуватись шляхо натиснення кнопки «ВХІД ШЕФ РЕЦ», буде виведено форму для авторизації (рис. 5.10)

Рисунок 5.10 – форма для авторизації

В разі неправильного логіну або паролю система виведе сповіщення про помилку ,якщо авторизація успішна буде виведено форму для планування завдань та замовлення запчастин (рис. 5.11)



Рисунок 5.11 – форма для планування завдань та замовлення запчастин

5.2 Випробування програмного продукту

В цьому підрозділі наведено опис тестів і порядок їх виконання для перевірки відповідності програмного забезпечення функціональним вимогам, зазначеним у технічному завданні на створення автоматизованої системи управління технічним обслуговуванням виробничого обладнання.

5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій автоматизованої системи управління технічним обслуговуванням обладнання вимогам вказаним у технічному завданні.

					ДП 6106.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

Продовження таблиці 5.1

№ випроб.	Ситуація (умова)	Ефект (дія)	Ефект фактичний
4	Користувач намагається переглянути реєстри обладнання або запчастин	Користувач натискає кнопку «ОБЛАДНАННЯ» після чого виводиться екранна форма перегляду реєстрів лотім користувач обирає тип реєстру шляхом натискання відповідної кнопки і в нижній частині форми виводяться дані з реєстру	Співпадає з очікуваним
5	Користувач редагує реєстр	Користувач вносить дані до полів (валідні) таблиці і натискає кнопку збереження потім закриває програму і знову відкриває через деякий час та переглядає відповідний реєстр : дані збережені	Співпадає з очікуваним
6	Користувач намагається знайти запчастину або обладнання	Користувач, заходить на форму перегляду реєстрів запчастин/обладнання і натискає кнопку пошуку виводиться форма для пошуку , користувач вносить до полів коректні дані, натискає кнопку пошуку результати пошуку виводяться в нижній частині форми перегляду реєстрів	Співпадає з очікуваним

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 5.1

№ випроб.	Ситуація (умова)	Ефект (дія)	Ефект фактичний
7	Користувач редагує реєстр	Користувач вносить неправильний тип даних до полів таблиці і отримує повідомлення про помилку	Співпадає з очікуваним
8	Користувач намагається знайти запчастину або обладнання	Користувач, заходить на форму перегляду реєстрів запчастин/обладнання і натискає кнопку пошуку, виводиться форма для пошуку, користувач вносить до полів не коректні дані виводиться повідомлення про помилку	Співпадає з очікуваним
9	Користувач намагається додати до реєстру запчастину/обладнання	Користувач натискає відповідну кнопку виводиться форма для додавання, користувач вносить валідні дані і натискає кнопку зберегти потім переглянути реєстр, внесена інформація відображається в реєстрі	Співпадає з очікуваним

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 5.1

№ випроб.	Ситуація (умова)	Ефект (дія)	Ефект фактичний
11	Користувач намагається провести планування робіт	Користувач авторизується як начальник РЕЦ та натисканням відповідної кнопки переходить на форму планування робіт, натисканням кнопки редагує список поточних робіт	Співпадає з очікуваним
12	Користувач намагається створити бланк замовлення	Користувач авторизується як начальник РЕЦ та натисканням відповідної кнопки переходить на форму планування робіт, натискає кнопку замовлення запчастин, виводиться екранна форма, що містить елемент, в якому відображається список запчастин, які потрібно замовити та містить кнопку генерації бланку замовлення, після натискання якої генерується текстовий файл замовлення	Співпадає з очікуваним

Змн.	Арк.	№ докум.	Підпис	Дата

Висновок до розділу

Розділ містить керівництво користувача, котре описує дії, які можуть виконувати різні типи користувачів при використанні інформаційної системи. Користувачі системи можуть виконувати наступні дії: переглядати та редагувати дані в реєстрах, проводити пошук запчастин, витратних матеріалів, або обладнання за реєстрами, створювати бланк замовлення запчастин, виконувати задачі з планування робіт, отримувати завдання на проведення планових робіт з технічного обслуговування та вносити відомості про їх виконання. В керівництві користувача детально описано роботу з екранними формами, функціонал, доступ до якого вони надають, наведено їх вигляд.

ЗАГАЛЬНІ ВИСНОВКИ

В результаті виконання дипломного проєкту була досліджена предметна область, складено перелік функціональних вимог до інформаційної системи, відповідно до яких, складено перелік вхідних даних, необхідних для виконання поставлених перед системою завдань. Визначено перелік та формат вихідних даних та наведено їх структуру. Проведено детальний опис процесів та функцій, які реалізовані в інформаційній системі за допомогою діаграм діяльності та варіантів використання.

Використовуючи дані, які було отримано в процесі аналізу предметної області було проведено формулювання задач розробки системи, визначено цілі, яких необхідно досягти за допомогою використання інформаційної системи.

Розробка програмного забезпечення проводилась з використанням наступних засобів: мови програмування C# фреймворку Microsoft.NET з розширенням Framework Entity Framework, технології WindowsForms, системи управління базами даних Microsoft SQL Server 2019 та середовища

					ДП 6106.00.000 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

розробки Visual Studio 2019. Надано детальний опис інструментів розробки та проведено огляд переваг вибраних інструментів над їхніми аналогами.

При виконанні дипломного проекту було розроблено архітектуру програмного продукту що дозволяє розширювати систему , відповідно до потреб підприємства. Розроблено модель бази даних, яка надає можливість зберігання та доступу до всіх даних, необхідних системі для виконання поставлених перед нею задач. Управління базою даних відбувається за допомогою СУБД Microsoft SQL Server 2019. Схема бази даних дозволяє розширювати функціональні можливості системи в перспективі.

Дипломний проєкт містить керівництво користувача з експлуатації автоматизованої системи управління технічним обслуговуванням виробничого обладнання. Надано детальний опис кроків необхідних для виконання всіх функцій системи з наведенням копій відповідних екранних форм. Проведено опис випробувань інформаційної системи на предмет відповідності висунутим функціональним вимогам до програмного продукту, за результатами яких, розроблена система вимогам відповідає.

					ДП 6106.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

Перелік посилань

- 1 Entity Framework – інструкція побудови зав’язків за допомогою бібліотеки Entity Framework [Електронний ресурс] – Режим доступу до ресурсу: <https://msdn.microsoft.com/en-us/data/jj713564.aspx>
- 2 .NET Framework – інформація про архітектуру програмної платформи[Електронний ресурс] – режим доступу до ресурсу: https://ru.wikipedia.org/wiki/.NET_Framework
- 3 ADO.NET – опис технології[Електронний ресурс] – режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/ADO.NET>
- 4 ASP.NET – сторінка бібліотеки ASP.NET [Електронний ресурс] – Режим доступу до ресурса: <http://www.asp.net/get-started>.
- 5 Багатокритеріальний вибір – опис методів [Електронний ресурс] – режим доступу до ресурсу : https://life-prog.ru/view_optimtzation.php?id=2
- 6 Документація по .NET Framework [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/dotnet/framework/>
- 7 Ендрю Троелсен мова програмування С# 5.0 і платформа .NET 4.5, 6-е вид. Київ «в. б. Вільямс», 2013 – 1312 ст.
- 8 Кривництво по SQL Server [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/sql/sql-server/tutorials-for-sql-server-2016?view=sql-server-ver15>

Додаток А

Тексти програмного коду

*Автоматизована система управління технічним обслуговуванням
виробничого обладнання*

(Найменування програми (документа))

DVD-R

(Вид носія даних)

90 арк, 255 Кб

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2020 року

					ДП 6106.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

```

using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Sclad
{
    static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new FormStart());
        }
    }
}

using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

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

        private void button1_Click(object sender, EventArgs e)
        {
            System.Data.OleDb.OleDbDataReader myReader;
            string CustomerString;
            OleDbConnection1.Open();
            myReader = OleDbCommand1.ExecuteReader();
            while (myReader.Read())
            {
                CustomerString = myReader[1].ToString() + " " + myReader[2].ToString()
+ " " + myReader[3].ToString();
                listBox1.Items.Add(CustomerString);
                // // Додати результат в список ListBox
            }
            myReader.Close(); OleDbConnection1.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Form2 Exercise2 = new Form2();
            Exercise2.Show();
        }

        private void Clear(object sender, MouseEventArgs e)
        {
            listBox1.Items.Clear();
        }
    }
}

```

```

    }

    private void Form1_Load(object sender, EventArgs e)
    {

    }

    private void button3_Click(object sender, EventArgs e)
    {
        Form4 G = new Form4();
        G.Show();
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Sclad
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            oleDbDataAdapter1.Fill(mySteelDataDataSet1.Запчасти);
            mySteelDataDataSet1.AcceptChanges();
        }

        // private void button3_Click(object sender, EventArgs e)
        // {
        //     oleDbDataAdapter1.Update(mySteelDataDataSet1.Запчасти);
        // }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        private void button3_Click(object sender, EventArgs e)
        {
            oleDbDataAdapter1.Update(mySteelDataDataSet1);
        }

        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {

        }

        private void Form2_Load(object sender, EventArgs e)
        {

```

```

    }

    private void dataGridView1_CellContentClick_1(object sender,
DataGridViewCellEventArgs e)
    {
    }

    private void button4_Click(object sender, EventArgs e)
    {
        Form3 Q = new Form3();
        Q.Show();
    }

    private void button5_Click(object sender, EventArgs e)
    {
        Form5 jk = new Form5();
        jk.Show();
    }
    private void button6_Click(object sender, EventArgs e)
    {
        OleDbDataAdapter1.Fill(mySteelDataDataSet1.Запчасти);
        mySteelDataDataSet1.AcceptChanges();
    }
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Sclad
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        private void Form3_Load(object sender, EventArgs e)
        {
        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Sclad
{
    public partial class Form4 : Form
    {
        public Form4()
        {
            InitializeComponent();
        }

        private void Form4_Load(object sender, EventArgs e)
        {
        }

        private void button3_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Sclad
{
    public partial class Form5 : Form
    {
        public Form5()
        {
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}

```

```

using System;
using System.Collections.Generic;

```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Sclad
{
    public partial class Form6 : Form
    {
        public Form6()
        {
            InitializeComponent();
        }

        private void label2_Click(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form7 NC = new Form7();
            NC.Show();
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Sclad
{
    public partial class Form7 : Form
    {
        public Form7()
        {
            InitializeComponent();
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Sclad
{

```

```

public partial class FormStart : Form
{
    public FormStart()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        Form2 W = new Form2();
        W.Show();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        Form1 T = new Form1();
        T.Show();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        Search sed = new Search();
        sed.Show();
    }

    private void button4_Click(object sender, EventArgs e)
    {

    }

    private void FormStart_Load(object sender, EventArgs e)
    {

    }

    private void button4_Click_1(object sender, EventArgs e)
    {
        Form6 PL = new Form6();
        PL.Show();
    }
}
}

```

```
namespace Sclad {
```

```
[global::System.Serializable()]
```

```
[global::System.ComponentModel.DesignerCategoryAttribute("code")]
```

```
[global::System.ComponentModel.ToolboxItem(true)]
```

```
[global::System.Xml.Serialization.XmlSchemaProviderAttribute("GetTypedDataSetSchema")]
```

```
[global::System.Xml.Serialization.XmlRootAttribute("MySTeelDataDataSet")]
```

```

[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.DataSet")]
public partial class MySteelDataDataSet : global::System.Data.DataSet {

    private ЗапчастиDataTable tableЗапчасти;

    private global::System.Data.SchemaSerializationMode _schemaSerializationMode =
global::System.Data.SchemaSerializationMode.IncludeSchema;

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataS
etGenerator", "15.0.0.0")]

    public MySteelDataDataSet() {

        this.BeginInit();

        this.InitClass();

        global::System.ComponentModel.CollectionChangeEventHandler
schemaChangedHandler = new
global::System.ComponentModel.CollectionChangeEventHandler(this.SchemaChanged);

        base.Tables.CollectionChanged += schemaChangedHandler;

        base.Relations.CollectionChanged += schemaChangedHandler;

        this.EndInit();

    }

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataS
etGenerator", "15.0.0.0")]

    protected MySteelDataDataSet(global::System.Runtime.Serialization.SerializationInfo
info, global::System.Runtime.Serialization.StreamingContext context) :

        base(info, context, false) {

            if ((this.IsBinarySerialized(info, context) == true)) {

```

```

        this.InitVars(false);

        global::System.ComponentModel.CollectionChangeEventHandler
schemaChangedHandler1 = new
global::System.ComponentModel.CollectionChangeEventHandler(this.SchemaChanged);

        this.Tables.CollectionChanged += schemaChangedHandler1;

        this.Relations.CollectionChanged += schemaChangedHandler1;

        return;
    }

    string strSchema = ((string)(info.GetValue("XmlSchema", typeof(string))));

    if ((this.DetermineSchemaSerializationMode(info, context) ==
global::System.Data.SchemaSerializationMode.IncludeSchema)) {

        global::System.Data.DataSet ds = new global::System.Data.DataSet();

        ds.ReadXmlSchema(new global::System.Xml.XmlTextReader(new
global::System.IO.StringReader(strSchema)));

        if ((ds.Tables["Запчасти"] != null)) {

            base.Tables.Add(new ЗапчастиDataTable(ds.Tables["Запчасти"]));

        }

        this.DataSetName = ds.DataSetName;

        this.Prefix = ds.Prefix;

        this.Namespace = ds.Namespace;

        this.Locale = ds.Locale;

        this.CaseSensitive = ds.CaseSensitive;

        this.EnforceConstraints = ds.EnforceConstraints;

        this.Merge(ds, false, global::System.Data.MissingSchemaAction.Add);

        this.InitVars();

    }

    else {

        this.ReadXmlSchema(new global::System.Xml.XmlTextReader(new
global::System.IO.StringReader(strSchema)));

    }

```

```

        this.GetSerializationData(info, context);

        global::System.ComponentModel.CollectionChangeEventHandler
schemaChangedHandler = new
global::System.ComponentModel.CollectionChangeEventHandler(this.SchemaChanged);

        base.Tables.CollectionChanged += schemaChangedHandler;

        this.Relations.CollectionChanged += schemaChangedHandler;
    }

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
[global::System.ComponentModel.Browsable(false)]
```

```
[global::System.ComponentModel.DesignerSerializationVisibility(global::System.ComponentModel.DesignerSerializationVisibility.Content)]
```

```

public ЗапчастиDataTable Запчасти {
    get {
        return this.tableЗапчасти;
    }
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
[global::System.ComponentModel.BrowsableAttribute(true)]
```

```
[global::System.ComponentModel.DesignerSerializationVisibilityAttribute(global::System.ComponentModel.DesignerSerializationVisibility.Visible)]
```

```

public override global::System.Data.SchemaSerializationMode SchemaSerializationMode {
    get {

```

```

        return this._schemaSerializationMode;
    }
    set {
        this._schemaSerializationMode = value;
    }
}

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

[global::System.ComponentModel.DesignerSerializationVisibilityAttribute(global::System.ComponentModel.DesignerSerializationVisibility.Hidden)]

```

public new global::System.Data.DataTableCollection Tables {
    get {
        return base.Tables;
    }
}

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

[global::System.ComponentModel.DesignerSerializationVisibilityAttribute(global::System.ComponentModel.DesignerSerializationVisibility.Hidden)]

```

public new global::System.Data.DataRelationCollection Relations {
    get {
        return base.Relations;
    }
}

```

```
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
protected override void InitializeDerivedDataSet() {  
    this.BeginInit();  
    this.InitClass();  
    this.EndInit();  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public override global::System.Data.DataSet Clone() {  
    MySteelDataDataSet cln = ((MySteelDataDataSet)(base.Clone()));  
    cln.InitVars();  
    cln.SchemaSerializationMode = this.SchemaSerializationMode;  
    return cln;  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
protected override bool ShouldSerializeTables() {  
    return false;  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
protected override bool ShouldSerializeRelations() {  
    return false;  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
protected override void ReadXmlSerializable(global::System.Xml.XmlReader reader) {  
    if ((this.DetermineSchemaSerializationMode(reader) ==  
global::System.Data.SchemaSerializationMode.IncludeSchema)) {  
        this.Reset();  
        global::System.Data.DataSet ds = new global::System.Data.DataSet();  
        ds.ReadXml(reader);  
        if ((ds.Tables["Запчасти"] != null)) {  
            base.Tables.Add(new ЗапчастиDataTable(ds.Tables["Запчасти"]));  
        }  
        this.DataSetName = ds.DataSetName;  
        this.Prefix = ds.Prefix;  
        this.Namespace = ds.Namespace;  
        this.Locale = ds.Locale;  
        this.CaseSensitive = ds.CaseSensitive;  
        this.EnforceConstraints = ds.EnforceConstraints;  
        this.Merge(ds, false, global::System.Data.MissingSchemaAction.Add);  
        this.InitVars();  
    }  
}
```

```

    }
    else {
        this.ReadXml(reader);
        this.InitVars();
    }
}

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

```

protected override global::System.Xml.Schema.XmlSchema GetSchemaSerializable() {
    global::System.IO.MemoryStream stream = new global::System.IO.MemoryStream();
    this.WriteXmlSchema(new global::System.Xml.XmlTextWriter(stream, null));
    stream.Position = 0;
    return global::System.Xml.Schema.XmlSchema.Read(new
global::System.Xml.XmlTextReader(stream), null);
}

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

```

internal void InitVars() {
    this.InitVars(true);
}

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

```

internal void InitVars(bool initTable) {
    this.tableЗапчасти = ((ЗапчастиDataTable)(base.Tables["Запчасти"]));
    if ((initTable == true)) {
        if ((this.tableЗапчасти != null)) {
            this.tableЗапчасти.InitVars();
        }
    }
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```

private void InitClass() {
    this.DataSetName = "MySTeelDataDataSet";
    this.Prefix = "";
    this.Namespace = "http://tempuri.org/MySTeelDataDataSet.xsd";
    this.EnforceConstraints = true;
    this.SchemaSerializationMode =
global::System.Data.SchemaSerializationMode.IncludeSchema;
    this.tableЗапчасти = new ЗапчастиDataTable();
    base.Tables.Add(this.tableЗапчасти);
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```

private bool ShouldSerializeЗапчасти() {
    return false;
}

```

```
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
private void SchemaChanged(object sender,  
global::System.ComponentModel.CollectionChangeEventArgs e) {  
  
    if ((e.Action == global::System.ComponentModel.CollectionChangeAction.Remove)) {  
  
        this.InitVars();  
  
    }  
  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public static global::System.Xml.Schema.XmlSchemaComplexType  
GetTypedDataSetSchema(global::System.Xml.Schema.XmlSchemaSet xs) {
```

```
    MySteelDataDataSet ds = new MySteelDataDataSet();  
  
    global::System.Xml.Schema.XmlSchemaComplexType type = new  
global::System.Xml.Schema.XmlSchemaComplexType();  
  
    global::System.Xml.Schema.XmlSchemaSequence sequence = new  
global::System.Xml.Schema.XmlSchemaSequence();  
  
    global::System.Xml.Schema.XmlSchemaAny any = new  
global::System.Xml.Schema.XmlSchemaAny();  
  
    any.Namespace = ds.Namespace;  
  
    sequence.Items.Add(any);  
  
    type.Particle = sequence;  
  
    global::System.Xml.Schema.XmlSchema dsSchema = ds.GetSchemaSerializable();  
  
    if (xs.Contains(dsSchema.TargetNamespace)) {  
  
        global::System.IO.MemoryStream s1 = new global::System.IO.MemoryStream();
```

```

global::System.IO.MemoryStream s2 = new global::System.IO.MemoryStream();
try {
    global::System.Xml.Schema.XmlSchema schema = null;
    dsSchema.Write(s1);
    for (global::System.Collections.IEnumerator schemas =
xs.Schemas(dsSchema.TargetNamespace).GetEnumerator(); schemas.MoveNext(); ) {
        schema = ((global::System.Xml.Schema.XmlSchema)(schemas.Current));
        s2.SetLength(0);
        schema.Write(s2);
        if ((s1.Length == s2.Length)) {
            s1.Position = 0;
            s2.Position = 0;
            for (; ((s1.Position != s1.Length)
                && (s1.ReadByte() == s2.ReadByte())); ) {
                ;
            }
            if ((s1.Position == s1.Length)) {
                return type;
            }
        }
    }
}
finally {
    if ((s1 != null)) {
        s1.Close();
    }
    if ((s2 != null)) {
        s2.Close();
    }
}

```

```

    }
}
xs.Add(dsSchema);
return type;
}

```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public delegate void ЗапчастиRowChangeEventHandler(object sender,
ЗапчастиRowChangeEvent e);
```

```
/// <summary>
```

```
///Represents the strongly named DataTable class.
```

```
///</summary>
```

```
[global::System.Serializable()]
```

```
[global::System.Xml.Serialization.XmlSchemaProviderAttribute("GetTypedTableSchema")]
```

```
public partial class ЗапчастиDataTable :
global::System.Data.TypedTableBase<ЗапчастиRow> {
```

```
private global::System.Data.DataColumn columnКодТовара;
```

```
private global::System.Data.DataColumn columnМарка;
```

```
private global::System.Data.DataColumn columnСерийныйНомер;
```

```
private global::System.Data.DataColumn columnНаСкладе;
```

```
private global::System.Data.DataColumn columnЦена;
```

```
private global::System.Data.DataColumn columnЗаказано;
```

```
private global::System.Data.DataColumn columnМинимальныйЗапас;
```

```
private global::System.Data.DataColumn columnВремяЗадержки;
```

```
private global::System.Data.DataColumn columnКодТипа;
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public ЗапчастиDataTable() {  
    this.TableName = "Запчасти";  
    this.BeginInit();  
    this.InitClass();  
    this.EndInit();  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
internal ЗапчастиDataTable(global::System.Data.DataTable table) {  
    this.TableName = table.TableName;  
    if ((table.CaseSensitive != table.DataSet.CaseSensitive)) {  
        this.CaseSensitive = table.CaseSensitive;  
    }  
    if ((table.Locale.ToString() != table.DataSet.Locale.ToString())) {
```

```

        this.Locale = table.Locale;
    }
    if ((table.Namespace != table.DataSet.Namespace)) {
        this.Namespace = table.Namespace;
    }
    this.Prefix = table.Prefix;
    this.MinimumCapacity = table.MinimumCapacity;
}

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

protected ЗапчастиDataTable(global::System.Runtime.Serialization.SerializationInfo info, global::System.Runtime.Serialization.StreamingContext context) :

```

        base(info, context) {
            this.InitVars();
        }

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

```

public global::System.Data.DataColumn КодТовараColumn {
    get {
        return this.columnКодТовара;
    }
}

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public global::System.Data.DataColumn МаркаColumn {  
    get {  
        return this.columnМарка;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public global::System.Data.DataColumn СерийныйНомерColumn {  
    get {  
        return this.columnСерийныйНомер;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public global::System.Data.DataColumn НаСкладеColumn {  
    get {  
        return this.columnНаСкладе;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public global::System.Data.DataColumn ЦенаColumn {  
    get {  
        return this.columnЦена;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public global::System.Data.DataColumn ЗаказаноColumn {  
    get {  
        return this.columnЗаказано;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public global::System.Data.DataColumn МинимальныйЗапасColumn {  
    get {  
        return this.columnМинимальныйЗапас;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public global::System.Data.DataColumn ВремяЗадержкиColumn {  
    get {  
        return this.columnВремяЗадержки;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public global::System.Data.DataColumn КодТипаColumn {  
    get {  
        return this.columnКодТипа;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
[global::System.ComponentModel.Browsable(false)]
```

```
public int Count {  
    get {  
        return this.Rows.Count;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public ЗапчастиRow this[int index] {  
    get {  
        return ((ЗапчастиRow)(this.Rows[index]));  
    }  
}
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public event ЗапчастиRowChangeEventHandler ЗапчастиRowChanging;
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public event ЗапчастиRowChangeEventHandler ЗапчастиRowChanged;
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public event ЗапчастиRowChangeEventHandler ЗапчастиRowDeleting;
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public event ЗапчастиRowChangeEventHandler ЗапчастиRowDeleted;
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public void AddЗапчастиRow(ЗапчастиRow row) {  
    this.Rows.Add(row);  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public ЗапчастиRow AddЗапчастиRow(string Марка, string СерийныйНомер, int  
НаСкладе, decimal Цена, int Заказано, int МинимальныйЗапас, string ВремяЗадержки, int  
КодТипа) {
```

```
    ЗапчастиRow rowЗапчастиRow = ((ЗапчастиRow)(this.NewRow()));
```

```
    object[] columnValuesArray = new object[] {
```

```
        null,
```

```
        Марка,
```

```
        СерийныйНомер,
```

```
        НаСкладе,
```

```
        Цена,
```

```
        Заказано,
```

```
        МинимальныйЗапас,
```

```
        ВремяЗадержки,
```

```
        КодТипа};
```

```
    rowЗапчастиRow.ItemArray = columnValuesArray;
```

```
    this.Rows.Add(rowЗапчастиRow);
```

```
    return rowЗапчастиRow;
```

```
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public ЗапчастиRow FindByКодТовара(int КодТовара) {  
    return ((ЗапчастиRow)(this.Rows.Find(new object[] {  
        КодТовара})));  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public override global::System.Data.DataTable Clone() {  
    ЗапчастиDataTable cln = ((ЗапчастиDataTable)(base.Clone()));  
    cln.InitVars();  
    return cln;  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
protected override global::System.Data.DataTable CreateInstance() {  
    return new ЗапчастиDataTable();  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```

internal void InitVars() {

    this.columnКодТовара = base.Columns["КодТовара"];

    this.columnМарка = base.Columns["Марка"];

    this.columnСерийныйНомер = base.Columns["СерийныйНомер"];

    this.columnНаСкладе = base.Columns["НаСкладе"];

    this.columnЦена = base.Columns["Цена"];

    this.columnЗаказано = base.Columns["Заказано"];

    this.columnМинимальныйЗапас = base.Columns["МинимальныйЗапас"];

    this.columnВремяЗадержки = base.Columns["ВремяЗадержки"];

    this.columnКодТипа = base.Columns["КодТипа"];

}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```

private void InitClass() {

    this.columnКодТовара = new global::System.Data.DataColumn("КодТовара",
typeof(int), null, global::System.Data.MappingType.Element);

    base.Columns.Add(this.columnКодТовара);

    this.columnМарка = new global::System.Data.DataColumn("Марка", typeof(string),
null, global::System.Data.MappingType.Element);

    base.Columns.Add(this.columnМарка);

    this.columnСерийныйНомер = new
global::System.Data.DataColumn("СерийныйНомер", typeof(string), null,
global::System.Data.MappingType.Element);

    base.Columns.Add(this.columnСерийныйНомер);

    this.columnНаСкладе = new global::System.Data.DataColumn("НаСкладе",
typeof(int), null, global::System.Data.MappingType.Element);

    base.Columns.Add(this.columnНаСкладе);

```

```

        this.columnЦена = new global::System.Data.DataColumn("Цена", typeof(decimal),
null, global::System.Data.MappingType.Element);

        base.Columns.Add(this.columnЦена);

        this.columnЗаказано = new global::System.Data.DataColumn("Заказано", typeof(int),
null, global::System.Data.MappingType.Element);

        base.Columns.Add(this.columnЗаказано);

        this.columnМинимальныйЗапас = new
global::System.Data.DataColumn("МинимальныйЗапас", typeof(int), null,
global::System.Data.MappingType.Element);

        base.Columns.Add(this.columnМинимальныйЗапас);

        this.columnВремяЗадержки = new
global::System.Data.DataColumn("ВремяЗадержки", typeof(string), null,
global::System.Data.MappingType.Element);

        base.Columns.Add(this.columnВремяЗадержки);

        this.columnКодТипа = new global::System.Data.DataColumn("КодТипа", typeof(int),
null, global::System.Data.MappingType.Element);

        base.Columns.Add(this.columnКодТипа);

        this.Constraints.Add(new global::System.Data.UniqueConstraint("Constraint1", new
global::System.Data.DataColumn[] {

                this.columnКодТовара }, true));

        this.columnКодТовара.AutoIncrement = true;

        this.columnКодТовара.AutoIncrementSeed = -1;

        this.columnКодТовара.AutoIncrementStep = -1;

        this.columnКодТовара.AllowDBNull = false;

        this.columnКодТовара.Unique = true;

        this.columnМарка.MaxLength = 50;

        this.columnСерийныйНомер.MaxLength = 50;

        this.columnВремяЗадержки.MaxLength = 30;

    }

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public ЗапчастиRow NewЗапчастиRow() {  
    return ((ЗапчастиRow)(this.NewRow()));  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
protected override global::System.Data.DataRow  
NewRowFromBuilder(global::System.Data.DataRowBuilder builder) {  
    return new ЗапчастиRow(builder);  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
protected override global::System.Type GetRowType() {  
    return typeof(ЗапчастиRow);  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
protected override void  
OnRowChanged(global::System.Data.DataRowChangeEventArgs e) {  
    base.OnRowChanged(e);  
    if ((this.ЗапчастиRowChanged != null)) {
```

```

        this.ЗапчастиRowChanged(this, new
ЗапчастиRowChangeEvent(((ЗапчастиRow)(e.Row)), e.Action));
    }
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```

protected override void
OnRowChanging(global::System.Data.DataRowChangeEventArgs e) {
    base.OnRowChanging(e);
    if ((this.ЗапчастиRowChanging != null)) {
        this.ЗапчастиRowChanging(this, new
ЗапчастиRowChangeEvent(((ЗапчастиRow)(e.Row)), e.Action));
    }
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```

protected override void OnRowDeleted(global::System.Data.DataRowChangeEventArgs
e) {
    base.OnRowDeleted(e);
    if ((this.ЗапчастиRowDeleted != null)) {
        this.ЗапчастиRowDeleted(this, new
ЗапчастиRowChangeEvent(((ЗапчастиRow)(e.Row)), e.Action));
    }
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
protected override void OnRowDeleting(global::System.Data.DataRowChangeEventArgs e) {  
    base.OnRowDeleting(e);  
    if ((this.ЗапчастиRowDeleting != null)) {  
        this.ЗапчастиRowDeleting(this, new  
ЗапчастиRowChangeEvent(((ЗапчастиRow)(e.Row)), e.Action));  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public void RemoveЗапчастиRow(ЗапчастиRow row) {  
    this.Rows.Remove(row);  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public static global::System.Xml.Schema.XmlSchemaComplexType  
GetTypedTableSchema(global::System.Xml.Schema.XmlSchemaSet xs) {  
    global::System.Xml.Schema.XmlSchemaComplexType type = new  
global::System.Xml.Schema.XmlSchemaComplexType();  
  
    global::System.Xml.Schema.XmlSchemaSequence sequence = new  
global::System.Xml.Schema.XmlSchemaSequence();  
  
    MySteelDataDataSet ds = new MySteelDataDataSet();  
  
    global::System.Xml.Schema.XmlSchemaAny any1 = new  
global::System.Xml.Schema.XmlSchemaAny();
```

```

any1.Namespace = "http://www.w3.org/2001/XMLSchema";

any1.MinOccurs = new decimal(0);

any1.MaxOccurs = decimal.MaxValue;

any1.ProcessContents =
global::System.Xml.Schema.XmlSchemaContentProcessing.Lax;

sequence.Items.Add(any1);

global::System.Xml.Schema.XmlSchemaAny any2 = new
global::System.Xml.Schema.XmlSchemaAny();

any2.Namespace = "urn:schemas-microsoft-com:xml-diffgram-v1";

any2.MinOccurs = new decimal(1);

any2.ProcessContents =
global::System.Xml.Schema.XmlSchemaContentProcessing.Lax;

sequence.Items.Add(any2);

global::System.Xml.Schema.XmlSchemaAttribute attribute1 = new
global::System.Xml.Schema.XmlSchemaAttribute();

attribute1.Name = "namespace";

attribute1.FixedValue = ds.Namespace;

type.Attributes.Add(attribute1);

global::System.Xml.Schema.XmlSchemaAttribute attribute2 = new
global::System.Xml.Schema.XmlSchemaAttribute();

attribute2.Name = "tableName";

attribute2.FixedValue = "ЗапчастиDataTable";

type.Attributes.Add(attribute2);

type.Particle = sequence;

global::System.Xml.Schema.XmlSchema dsSchema = ds.GetSchemaSerializable();

if (xs.Contains(dsSchema.TargetNamespace)) {

    global::System.IO.MemoryStream s1 = new global::System.IO.MemoryStream();

    global::System.IO.MemoryStream s2 = new global::System.IO.MemoryStream();

    try {

        global::System.Xml.Schema.XmlSchema schema = null;

```

```

dsSchema.Write(s1);

for (global::System.Collections.IEnumerator schemas =
xs.Schemas(dsSchema.TargetNamespace).GetEnumerator(); schemas.MoveNext(); ) {

    schema = ((global::System.Xml.Schema.XmlSchema)(schemas.Current));

    s2.SetLength(0);

    schema.Write(s2);

    if ((s1.Length == s2.Length) {

        s1.Position = 0;

        s2.Position = 0;

        for (; ((s1.Position != s1.Length)

            && (s1.ReadByte() == s2.ReadByte())); ) {

            ;

        }

        if ((s1.Position == s1.Length) {

            return type;

        }

    }

}

}

finally {

    if ((s1 != null) {

        s1.Close();

    }

    if ((s2 != null) {

        s2.Close();

    }

}

}

xs.Add(dsSchema);

```

```

        return type;
    }
}

/// <summary>
/// Represents strongly named DataRow class.
/// </summary>
public partial class ЗапчастиRow : global::System.Data.DataRow {

    private ЗапчастиDataTable tableЗапчасти;

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

    internal ЗапчастиRow(global::System.Data.DataRowBuilder rb) :
        base(rb) {
        this.tableЗапчасти = ((ЗапчастиDataTable)(this.Table));
    }

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

    public int КодТовара {
        get {
            return ((int)(this[this.tableЗапчасти.КодТовараColumn]));
        }
        set {
            this[this.tableЗапчасти.КодТовараColumn] = value;
        }
    }
}

```

```
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public string Марка {  
    get {  
        try {  
            return ((string)(this[this.tableЗапчасти.МаркаColumn]));  
        }  
        catch (global::System.InvalidCastException e) {  
            throw new global::System.Data.StrongTypingException("Значение для столбца  
'Марка' в таблице 'Запчасти' равно DBNull.", e);  
        }  
    }  
    set {  
        this[this.tableЗапчасти.МаркаColumn] = value;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public string СерийныйНомер {  
    get {  
        try {  
            return ((string)(this[this.tableЗапчасти.СерийныйНомерColumn]));  
        }  
    }  
}
```

```

    }

    catch (global::System.InvalidCastException e) {

        throw new global::System.Data.StrongTypingException("Значение для столбца
        \СерийныйНомер\ в таблице \Запчасти\ равно DBNull.", e);

    }

}

set {

    this[this.tableЗапчасти.СерийныйНомерColumn] = value;

}

}

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

```

public int НаСкладе {

    get {

        try {

            return ((int)(this[this.tableЗапчасти.НаСкладеColumn]));

        }

        catch (global::System.InvalidCastException e) {

            throw new global::System.Data.StrongTypingException("Значение для столбца
            \НаСкладе\ в таблице \Запчасти\ равно DBNull.", e);

        }

    }

    set {

        this[this.tableЗапчасти.НаСкладеColumn] = value;

    }

}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public decimal Цена {  
    get {  
        try {  
            return ((decimal)(this[this.tableЗапчасти.ЦенаColumn]));  
        }  
        catch (global::System.InvalidCastException e) {  
            throw new global::System.Data.StrongTypingException("Значение для столбца  
'Цена' в таблице 'Запчасти' равно DBNull.", e);  
        }  
    }  
    set {  
        this[this.tableЗапчасти.ЦенаColumn] = value;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public int Заказано {  
    get {  
        try {  
            return ((int)(this[this.tableЗапчасти.ЗаказаноColumn]));  
        }  
        catch (global::System.InvalidCastException e) {
```

```

        throw new global::System.Data.StrongTypingException("Значение для столбца
\Заказано\ в таблице \Запчасти\ равно DBNull.", e);
    }
}
set {
    this[this.tableЗапчасти.ЗаказаноColumn] = value;
}
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```

public int МинимальныйЗапас {
    get {
        try {
            return ((int)(this[this.tableЗапчасти.МинимальныйЗапасColumn]));
        }
        catch (global::System.InvalidCastException e) {
            throw new global::System.Data.StrongTypingException("Значение для столбца
\МинимальныйЗапас\ в таблице \Запчасти\ равно DBNull.", e);
        }
    }
    set {
        this[this.tableЗапчасти.МинимальныйЗапасColumn] = value;
    }
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public string ВремяЗадержки {  
    get {  
        try {  
            return ((string)(this[this.tableЗапчасти.ВремяЗадержкиColumn]));  
        }  
        catch (global::System.InvalidCastException e) {  
            throw new global::System.Data.StrongTypingException("Значение для столбца  
'ВремяЗадержки' в таблице 'Запчасти' равно DBNull.", e);  
        }  
    }  
    set {  
        this[this.tableЗапчасти.ВремяЗадержкиColumn] = value;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public int КодТипа {  
    get {  
        try {  
            return ((int)(this[this.tableЗапчасти.КодТипаColumn]));  
        }  
        catch (global::System.InvalidCastException e) {  
            throw new global::System.Data.StrongTypingException("Значение для столбца  
'КодТипа' в таблице 'Запчасти' равно DBNull.", e);  
        }  
    }  
}
```

```
    }  
  }  
  set {  
    this[this.tableЗапчасти.КодТипаColumn] = value;  
  }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public bool IsМаркаNull() {  
    return this.IsNull(this.tableЗапчасти.МаркаColumn);  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public void SetМаркаNull() {  
    this[this.tableЗапчасти.МаркаColumn] = global::System.Convert.DBNull;  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public bool IsСерийныйНомерNull() {  
    return this.IsNull(this.tableЗапчасти.СерийныйНомерColumn);  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public void SetСерийныйНомерNull() {  
    this[this.tableЗапчасти.СерийныйНомерColumn] = global::System.Convert.DBNull;  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public bool IsНаСкладеNull() {  
    return this.IsNull(this.tableЗапчасти.НаСкладеColumn);  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public void SetНаСкладеNull() {  
    this[this.tableЗапчасти.НаСкладеColumn] = global::System.Convert.DBNull;  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public bool IsЦенаNull() {  
    return this.IsNull(this.tableЗапчасти.ЦенаColumn);  
}
```

```
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public void SetЦенаNull() {  
    this[this.tableЗапчасти.ЦенаColumn] = global::System.Convert.DBNull;  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public bool IsЗаказаноNull() {  
    return this.IsNull(this.tableЗапчасти.ЗаказаноColumn);  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public void SetЗаказаноNull() {  
    this[this.tableЗапчасти.ЗаказаноColumn] = global::System.Convert.DBNull;  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public bool IsМинимальныйЗапасNull() {
```

```
        return this.IsNull(this.tableЗапчасти.МинимальныйЗапасColumn);  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
    public void SetМинимальныйЗапасNull() {  
        this[this.tableЗапчасти.МинимальныйЗапасColumn] =  
global::System.Convert.DBNull;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
    public bool IsВремяЗадержкиNull() {  
        return this.IsNull(this.tableЗапчасти.ВремяЗадержкиColumn);  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
    public void SetВремяЗадержкиNull() {  
        this[this.tableЗапчасти.ВремяЗадержкиColumn] = global::System.Convert.DBNull;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public bool IsКодТипаNull() {  
    return this.IsNull(this.tableЗапчасти.КодТипаColumn);  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public void SetКодТипаNull() {  
    this[this.tableЗапчасти.КодТипаColumn] = global::System.Convert.DBNull;  
}  
}
```

```
/// <summary>
```

```
///Row event argument class
```

```
///</summary>
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public class ЗапчастиRowChangeEvent : global::System.EventArgs {
```

```
    private ЗапчастиRow eventRow;
```

```
    private global::System.Data.DataRowAction eventAction;
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
    public ЗапчастиRowChangeEvent(ЗапчастиRow row,  
    global::System.Data.DataRowAction action) {
```

```
    this.eventRow = row;

    this.eventAction = action;
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

```

```
public ЗапчастиRow Row {
    get {
        return this.eventRow;
    }
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

```

```
public global::System.Data.DataRowAction Action {
    get {
        return this.eventAction;
    }
}
}
}
}
}

```

```
namespace Sclad.MySteelDataDataSetTableAdapters {

```

```
/// <summary>
```

```

//Represents the connection and commands used to retrieve and save data.

//</summary>

[global::System.ComponentModel.DesignerCategoryAttribute("code")]

[global::System.ComponentModel.ToolboxItem(true)]

[global::System.ComponentModel.DataObjectAttribute(true)]

[global::System.ComponentModel.DesignerAttribute("Microsoft.VSDesigner.DataSource.Design.TableAdapterDesigner, Microsoft.VSDesigner" +
    ", Version=10.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")]

[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public partial class ЗапчастиTableAdapter : global::System.ComponentModel.Component {

    private global::System.Data.OleDb.OleDbDataAdapter _adapter;

    private global::System.Data.OleDb.OleDbConnection _connection;

    private global::System.Data.OleDb.OleDbTransaction _transaction;

    private global::System.Data.OleDb.OleDbCommand[] _commandCollection;

    private bool _clearBeforeFill;

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

    public ЗапчастиTableAdapter() {

        this.ClearBeforeFill = true;

    }

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
protected internal global::System.Data.OleDb.OleDbDataAdapter Adapter {  
    get {  
        if ((this._adapter == null)) {  
            this.InitAdapter();  
        }  
        return this._adapter;  
    }  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
internal global::System.Data.OleDb.OleDbConnection Connection {  
    get {  
        if ((this._connection == null)) {  
            this.InitConnection();  
        }  
        return this._connection;  
    }  
    set {  
        this._connection = value;  
        if ((this.Adapter.InsertCommand != null)) {  
            this.Adapter.InsertCommand.Connection = value;  
        }  
        if ((this.Adapter.DeleteCommand != null)) {
```

```

        this.Adapter.DeleteCommand.Connection = value;
    }
    if ((this.Adapter.UpdateCommand != null)) {
        this.Adapter.UpdateCommand.Connection = value;
    }
    for (int i = 0; (i < this.CommandCollection.Length); i = (i + 1)) {
        if ((this.CommandCollection[i] != null)) {
            ((global::System.Data.OleDb.OleDbCommand)(this.CommandCollection[i])).Connection =
            value;
        }
    }
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```

internal global::System.Data.OleDb.OleDbTransaction Transaction {
    get {
        return this._transaction;
    }
    set {
        this._transaction = value;
        for (int i = 0; (i < this.CommandCollection.Length); i = (i + 1)) {
            this.CommandCollection[i].Transaction = this._transaction;
        }
        if (((this.Adapter != null)
            && (this.Adapter.DeleteCommand != null))) {

```

```

        this.Adapter.DeleteCommand.Transaction = this._transaction;
    }
    if (((this.Adapter != null)
        && (this.Adapter.InsertCommand != null))) {
        this.Adapter.InsertCommand.Transaction = this._transaction;
    }
    if (((this.Adapter != null)
        && (this.Adapter.UpdateCommand != null))) {
        this.Adapter.UpdateCommand.Transaction = this._transaction;
    }
}
}
}

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

```

protected global::System.Data.OleDb.OleDbCommand[] CommandCollection {
    get {
        if ((this._commandCollection == null)) {
            this.InitCommandCollection();
        }
        return this._commandCollection;
    }
}
}

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

```

public bool ClearBeforeFill {
    get {
        return this._clearBeforeFill;
    }
    set {
        this._clearBeforeFill = value;
    }
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```

private void InitAdapter() {
    this._adapter = new global::System.Data.OleDb.OleDbDataAdapter();
    global::System.Data.Common.DataTableMapping tableMapping = new
global::System.Data.Common.DataTableMapping();
    tableMapping.SourceTable = "Table";
    tableMapping.DataSetTable = "Запчасти";
    tableMapping.ColumnMappings.Add("КодТовара", "КодТовара");
    tableMapping.ColumnMappings.Add("Марка", "Марка");
    tableMapping.ColumnMappings.Add("СерийныйНомер", "СерийныйНомер");
    tableMapping.ColumnMappings.Add("НаСкладе", "НаСкладе");
    tableMapping.ColumnMappings.Add("Цена", "Цена");
    tableMapping.ColumnMappings.Add("Заказано", "Заказано");
    tableMapping.ColumnMappings.Add("МинимальныйЗапас", "МинимальныйЗапас");
    tableMapping.ColumnMappings.Add("ВремяЗадержки", "ВремяЗадержки");
    tableMapping.ColumnMappings.Add("КодТипа", "КодТипа");
    this._adapter.TableMappings.Add(tableMapping);
}

```

```

this._adapter.DeleteCommand = new global::System.Data.OleDb.OleDbCommand();

this._adapter.DeleteCommand.Connection = this.Connection;

this._adapter.DeleteCommand.CommandText = @"DELETE FROM `Запчасти`
WHERE ((`КодТовара` = ?) AND ((? = 1 AND `Марка` IS NULL) OR (`Марка` = ?)) AND
((? = 1 AND `СерийныйНомер` IS NULL) OR (`СерийныйНомер` = ?)) AND ((? = 1 AND
`НаСкладе` IS NULL) OR (`НаСкладе` = ?)) AND ((? = 1 AND `Цена` IS NULL) OR (`Цена`
= ?)) AND ((? = 1 AND `Заказано` IS NULL) OR (`Заказано` = ?)) AND ((? = 1 AND
`МинимальныйЗапас` IS NULL) OR (`МинимальныйЗапас` = ?)) AND ((? = 1 AND
`ВремяЗадержки` IS NULL) OR (`ВремяЗадержки` = ?)) AND ((? = 1 AND `КодТипа` IS
NULL) OR (`КодТипа` = ?))";

this._adapter.DeleteCommand.CommandType =
global::System.Data.CommandType.Text;

this._adapter.DeleteCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Original_КодТовара",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "КодТовара",
global::System.Data.DataRowVersion.Original, false, null));

this._adapter.DeleteCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("IsNull_Марка",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Марка",
global::System.Data.DataRowVersion.Original, true, null));

this._adapter.DeleteCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Original_Марка",
global::System.Data.OleDb.OleDbType.VarWChar, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Марка",
global::System.Data.DataRowVersion.Original, false, null));

this._adapter.DeleteCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("IsNull_СерийныйНомер",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "СерийныйНомер",
global::System.Data.DataRowVersion.Original, true, null));

this._adapter.DeleteCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Original_СерийныйНомер",
global::System.Data.OleDb.OleDbType.VarWChar, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "СерийныйНомер",
global::System.Data.DataRowVersion.Original, false, null));

this._adapter.DeleteCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("IsNull_НаСкладе",
global::System.Data.OleDb.OleDbType.Integer, 0,

```

```
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "HaСкладе",  
global::System.Data.DataRowVersion.Original, true, null));
```

```
    this._adapter.DeleteCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("Original_HaСкладе",  
global::System.Data.OleDb.OleDbType.Integer, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "HaСкладе",  
global::System.Data.DataRowVersion.Original, false, null));
```

```
    this._adapter.DeleteCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("IsNull_Цена",  
global::System.Data.OleDb.OleDbType.Integer, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Цена",  
global::System.Data.DataRowVersion.Original, true, null));
```

```
    this._adapter.DeleteCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("Original_Цена",  
global::System.Data.OleDb.OleDbType.Currency, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Цена",  
global::System.Data.DataRowVersion.Original, false, null));
```

```
    this._adapter.DeleteCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("IsNull_Заказано",  
global::System.Data.OleDb.OleDbType.Integer, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Заказано",  
global::System.Data.DataRowVersion.Original, true, null));
```

```
    this._adapter.DeleteCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("Original_Заказано",  
global::System.Data.OleDb.OleDbType.Integer, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Заказано",  
global::System.Data.DataRowVersion.Original, false, null));
```

```
    this._adapter.DeleteCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("IsNull_МинимальныйЗапас",  
global::System.Data.OleDb.OleDbType.Integer, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "МинимальныйЗапас",  
global::System.Data.DataRowVersion.Original, true, null));
```

```
    this._adapter.DeleteCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("Original_МинимальныйЗапас",  
global::System.Data.OleDb.OleDbType.Integer, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "МинимальныйЗапас",  
global::System.Data.DataRowVersion.Original, false, null));
```

```
    this._adapter.DeleteCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("IsNull_ВремяЗадержки",  
global::System.Data.OleDb.OleDbType.Integer, 0,
```

```

global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "ВремяЗадержки",
global::System.Data.DataRowVersion.Original, true, null));

        this._adapter.DeleteCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Original_ВремяЗадержки",
global::System.Data.OleDb.OleDbType.VarWChar, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "ВремяЗадержки",
global::System.Data.DataRowVersion.Original, false, null));

        this._adapter.DeleteCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("IsNull_КодТипа",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "КодТипа",
global::System.Data.DataRowVersion.Original, true, null));

        this._adapter.DeleteCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Original_КодТипа",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "КодТипа",
global::System.Data.DataRowVersion.Original, false, null));

        this._adapter.InsertCommand = new global::System.Data.OleDb.OleDbCommand();

        this._adapter.InsertCommand.Connection = this.Connection;

        this._adapter.InsertCommand.CommandText = "INSERT INTO `Запчасти` (`Марка`,
`СерийныйНомер`, `НаСкладе`, `Цена`, `Заказано`, " +
            " `МинимальныйЗапас`, `ВремяЗадержки`, `КодТипа`) VALUES (?, ?, ?, ?, ?, ?, ?,
?)" +
            """;

        this._adapter.InsertCommand.CommandType =
global::System.Data.CommandType.Text;

        this._adapter.InsertCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Марка",
global::System.Data.OleDb.OleDbType.VarWChar, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Марка",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.InsertCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("СерийныйНомер",
global::System.Data.OleDb.OleDbType.VarWChar, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "СерийныйНомер",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.InsertCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("НаСкладе",

```

```

global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "НаСкладе",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.InsertCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Цена",
global::System.Data.OleDb.OleDbType.Currency, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Цена",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.InsertCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Заказано",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Заказано",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.InsertCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("МинимальныйЗапас",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "МинимальныйЗапас",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.InsertCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("ВремяЗадержки",
global::System.Data.OleDb.OleDbType.VarWChar, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "ВремяЗадержки",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.InsertCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("КодТипа",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "КодТипа",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.UpdateCommand = new global::System.Data.OleDb.OleDbCommand();

        this._adapter.UpdateCommand.Connection = this.Connection;

        this._adapter.UpdateCommand.CommandText = @"UPDATE `Запчасти` SET `Марка`
= ?, `СерийныйНомер` = ?, `НаСкладе` = ?, `Цена` = ?, `Заказано` = ?,
`МинимальныйЗапас` = ?, `ВремяЗадержки` = ?, `КодТипа` = ? WHERE ((`КодТовара` = ?)
AND ((? = 1 AND `Марка` IS NULL) OR (`Марка` = ?)) AND ((? = 1 AND
`СерийныйНомер` IS NULL) OR (`СерийныйНомер` = ?)) AND ((? = 1 AND `НаСкладе` IS
NULL) OR (`НаСкладе` = ?)) AND ((? = 1 AND `Цена` IS NULL) OR (`Цена` = ?)) AND ((?
= 1 AND `Заказано` IS NULL) OR (`Заказано` = ?)) AND ((? = 1 AND `МинимальныйЗапас`
IS NULL) OR (`МинимальныйЗапас` = ?)) AND ((? = 1 AND `ВремяЗадержки` IS NULL)
OR (`ВремяЗадержки` = ?)) AND ((? = 1 AND `КодТипа` IS NULL) OR (`КодТипа` = ?))";

```

```

        this._adapter.UpdateCommand.CommandType =
global::System.Data.CommandType.Text;

        this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Марка",
global::System.Data.OleDb.OleDbType.VarWChar, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Марка",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("СерийныйНомер",
global::System.Data.OleDb.OleDbType.VarWChar, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "СерийныйНомер",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("НаСкладе",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "НаСкладе",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Цена",
global::System.Data.OleDb.OleDbType.Currency, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Цена",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Заказано",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Заказано",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("МинимальныйЗапас",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "МинимальныйЗапас",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("ВремяЗадержки",
global::System.Data.OleDb.OleDbType.VarWChar, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "ВремяЗадержки",
global::System.Data.DataRowVersion.Current, false, null));

        this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("КодТипа",
global::System.Data.OleDb.OleDbType.Integer, 0,

```

```
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "КодТипа",  
global::System.Data.DataRowVersion.Current, false, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("Original_КодТовара",  
global::System.Data.OleDb.OleDbType.Integer, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "КодТовара",  
global::System.Data.DataRowVersion.Original, false, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("IsNull_Марка",  
global::System.Data.OleDb.OleDbType.Integer, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Марка",  
global::System.Data.DataRowVersion.Original, true, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("Original_Марка",  
global::System.Data.OleDb.OleDbType.VarWChar, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Марка",  
global::System.Data.DataRowVersion.Original, false, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("IsNull_СерийныйНомер",  
global::System.Data.OleDb.OleDbType.Integer, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "СерийныйНомер",  
global::System.Data.DataRowVersion.Original, true, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("Original_СерийныйНомер",  
global::System.Data.OleDb.OleDbType.VarWChar, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "СерийныйНомер",  
global::System.Data.DataRowVersion.Original, false, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("IsNull_НаСкладе",  
global::System.Data.OleDb.OleDbType.Integer, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "НаСкладе",  
global::System.Data.DataRowVersion.Original, true, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("Original_НаСкладе",  
global::System.Data.OleDb.OleDbType.Integer, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "НаСкладе",  
global::System.Data.DataRowVersion.Original, false, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("IsNull_Цена",  
global::System.Data.OleDb.OleDbType.Integer, 0,
```

```
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Цена",
global::System.Data.DataRowVersion.Original, true, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Original_Цена",
global::System.Data.OleDb.OleDbType.Currency, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Цена",
global::System.Data.DataRowVersion.Original, false, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("IsNull_Заказано",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Заказано",
global::System.Data.DataRowVersion.Original, true, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Original_Заказано",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "Заказано",
global::System.Data.DataRowVersion.Original, false, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("IsNull_МинимальныйЗапас",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "МинимальныйЗапас",
global::System.Data.DataRowVersion.Original, true, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Original_МинимальныйЗапас",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "МинимальныйЗапас",
global::System.Data.DataRowVersion.Original, false, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("IsNull_ВремяЗадержки",
global::System.Data.OleDb.OleDbType.Integer, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "ВремяЗадержки",
global::System.Data.DataRowVersion.Original, true, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("Original_ВремяЗадержки",
global::System.Data.OleDb.OleDbType.VarWChar, 0,
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "ВремяЗадержки",
global::System.Data.DataRowVersion.Original, false, null));
```

```
    this._adapter.UpdateCommand.Parameters.Add(new
global::System.Data.OleDb.OleDbParameter("IsNull_КодТипа",
global::System.Data.OleDb.OleDbType.Integer, 0,
```

```
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "КодТипа",  
global::System.Data.DataRowVersion.Original, true, null));
```

```
        this._adapter.UpdateCommand.Parameters.Add(new  
global::System.Data.OleDb.OleDbParameter("Original_КодТипа",  
global::System.Data.OleDb.OleDbType.Integer, 0,  
global::System.Data.ParameterDirection.Input, ((byte)(0)), ((byte)(0)), "КодТипа",  
global::System.Data.DataRowVersion.Original, false, null));
```

```
    }
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataS  
etGenerator", "15.0.0.0")]
```

```
private void InitConnection() {
```

```
    this._connection = new global::System.Data.OleDb.OleDbConnection();
```

```
    this._connection.ConnectionString =  
global::Scld.Properties.Settings.Default.MySTeelDataConnectionString;
```

```
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataS  
etGenerator", "15.0.0.0")]
```

```
private void InitCommandCollection() {
```

```
    this._commandCollection = new global::System.Data.OleDb.OleDbCommand[1];
```

```
    this._commandCollection[0] = new global::System.Data.OleDb.OleDbCommand();
```

```
    this._commandCollection[0].Connection = this.Connection;
```

```
    this._commandCollection[0].CommandText = "SELECT КодТовара, Марка,  
СерийныйНомер, НаСкладе, Цена, Заказано, МинимальныйЗапа" +
```

```
        "с, ВремяЗадержки, КодТипа FROM Запчасти";
```

```
    this._commandCollection[0].CommandType = global::System.Data.CommandType.Text;
```

```
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]
```

```
[global::System.ComponentModel.DataObjectMethodAttribute(global::System.ComponentModel.DataObjectMethodType.Fill, true)]
```

```
public virtual int Fill(MySTeelDataDataSet.ЗапчастиDataTable dataTable) {  
    this.Adapter.SelectCommand = this.CommandCollection[0];  
    if ((this.ClearBeforeFill == true)) {  
        dataTable.Clear();  
    }  
    int returnValue = this.Adapter.Fill(dataTable);  
    return returnValue;  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]
```

```
[global::System.ComponentModel.DataObjectMethodAttribute(global::System.ComponentModel.DataObjectMethodType.Select, true)]
```

```
public virtual MySTeelDataDataSet.ЗапчастиDataTable GetData() {  
    this.Adapter.SelectCommand = this.CommandCollection[0];  
    MySTeelDataDataSet.ЗапчастиDataTable dataTable = new  
MySTeelDataDataSet.ЗапчастиDataTable();  
    this.Adapter.Fill(dataTable);  
    return dataTable;  
}
```

```
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]
```

```
public virtual int Update(MySteelDataDataSet.ЗапчастиDataTable dataTable) {
```

```
    return this.Adapter.Update(dataTable);
```

```
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]
```

```
public virtual int Update(MySteelDataDataSet dataSet) {
```

```
    return this.Adapter.Update(dataSet, "Запчасти");
```

```
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]
```

```
public virtual int Update(global::System.Data.DataRow dataRow) {
```

```
    return this.Adapter.Update(new global::System.Data.DataRow[] {  
        dataRow});
```

```
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]
```

```
public virtual int Update(global::System.Data.DataRow[] dataRows) {  
    return this.Adapter.Update(dataRows);  
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]
```

```
[global::System.ComponentModel.DataObjectMethodAttribute(global::System.ComponentModel.DataObjectMethodType.Delete, true)]
```

```
public virtual int Delete(int Original_КодТовара, string Original_Марка, string  
Original_СерийныйНомер, global::System.Nullable<int> Original_НаСкладе,  
global::System.Nullable<decimal> Original_Цена, global::System.Nullable<int>  
Original_Заказано, global::System.Nullable<int> Original_МинимальныйЗапас, string  
Original_ВремяЗадержки, global::System.Nullable<int> Original_КодТипа) {
```

```
    this.Adapter.DeleteCommand.Parameters[0].Value = ((int)(Original_КодТовара));
```

```
    if ((Original_Марка == null)) {
```

```
        this.Adapter.DeleteCommand.Parameters[1].Value = ((object)(1));
```

```
        this.Adapter.DeleteCommand.Parameters[2].Value = global::System.DBNull.Value;
```

```
    }
```

```
    else {
```

```
        this.Adapter.DeleteCommand.Parameters[1].Value = ((object)(0));
```

```
        this.Adapter.DeleteCommand.Parameters[2].Value = ((string)(Original_Марка));
```

```
    }
```

```
    if ((Original_СерийныйНомер == null)) {
```

```

this.Adapter.DeleteCommand.Parameters[3].Value = ((object)(1));

this.Adapter.DeleteCommand.Parameters[4].Value = global::System.DBNull.Value;
}

else {

    this.Adapter.DeleteCommand.Parameters[3].Value = ((object)(0));

    this.Adapter.DeleteCommand.Parameters[4].Value =
((string)(Original_СерийныйНомер));
}

if ((Original_НаСкладе.HasValue == true)) {

    this.Adapter.DeleteCommand.Parameters[5].Value = ((object)(0));

    this.Adapter.DeleteCommand.Parameters[6].Value =
((int)(Original_НаСкладе.Value));
}

else {

    this.Adapter.DeleteCommand.Parameters[5].Value = ((object)(1));

    this.Adapter.DeleteCommand.Parameters[6].Value = global::System.DBNull.Value;
}

if ((Original_Цена.HasValue == true)) {

    this.Adapter.DeleteCommand.Parameters[7].Value = ((object)(0));

    this.Adapter.DeleteCommand.Parameters[8].Value =
((decimal)(Original_Цена.Value));
}

else {

    this.Adapter.DeleteCommand.Parameters[7].Value = ((object)(1));

    this.Adapter.DeleteCommand.Parameters[8].Value = global::System.DBNull.Value;
}

if ((Original_Заказано.HasValue == true)) {

    this.Adapter.DeleteCommand.Parameters[9].Value = ((object)(0));

    this.Adapter.DeleteCommand.Parameters[10].Value =
((int)(Original_Заказано.Value));
}

```

```

}
else {
    this.Adapter.DeleteCommand.Parameters[9].Value = ((object)(1));
    this.Adapter.DeleteCommand.Parameters[10].Value = global::System.DBNull.Value;
}
if ((Original_МинимальныйЗапас.HasValue == true)) {
    this.Adapter.DeleteCommand.Parameters[11].Value = ((object)(0));
    this.Adapter.DeleteCommand.Parameters[12].Value =
((int)(Original_МинимальныйЗапас.Value));
}
else {
    this.Adapter.DeleteCommand.Parameters[11].Value = ((object)(1));
    this.Adapter.DeleteCommand.Parameters[12].Value = global::System.DBNull.Value;
}
if ((Original_ВремяЗадержки == null)) {
    this.Adapter.DeleteCommand.Parameters[13].Value = ((object)(1));
    this.Adapter.DeleteCommand.Parameters[14].Value = global::System.DBNull.Value;
}
else {
    this.Adapter.DeleteCommand.Parameters[13].Value = ((object)(0));
    this.Adapter.DeleteCommand.Parameters[14].Value =
((string)(Original_ВремяЗадержки));
}
if ((Original_КодТипа.HasValue == true)) {
    this.Adapter.DeleteCommand.Parameters[15].Value = ((object)(0));
    this.Adapter.DeleteCommand.Parameters[16].Value =
((int)(Original_КодТипа.Value));
}
else {

```

```

        this.Adapter.DeleteCommand.Parameters[15].Value = ((object)(1));

        this.Adapter.DeleteCommand.Parameters[16].Value = global::System.DBNull.Value;
    }

    global::System.Data.ConnectionState previousConnectionState =
this.Adapter.DeleteCommand.Connection.State;

    if (((this.Adapter.DeleteCommand.Connection.State &
global::System.Data.ConnectionState.Open)

        != global::System.Data.ConnectionState.Open)) {

        this.Adapter.DeleteCommand.Connection.Open();
    }

    try {

        int returnValue = this.Adapter.DeleteCommand.ExecuteNonQuery();

        return returnValue;
    }

    finally {

        if ((previousConnectionState == global::System.Data.ConnectionState.Closed)) {

            this.Adapter.DeleteCommand.Connection.Close();

        }

    }

}

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[global::System.ComponentModel.DataObjectMethodAttribute(global::System.ComponentModel.DataObjectMethodType.Insert, true)]

```

    public virtual int Insert(string Марка, string СерийныйНомер,
global::System.Nullable<int> НаСкладе, global::System.Nullable<decimal> Цена,

```

```

global::System.Nullable<int> Заказано, global::System.Nullable<int> МинимальныйЗапас,
string ВремяЗадержки, global::System.Nullable<int> КодТипа) {
    if ((Марка == null)) {
        this.Adapter.InsertCommand.Parameters[0].Value = global::System.DBNull.Value;
    }
    else {
        this.Adapter.InsertCommand.Parameters[0].Value = ((string)(Марка));
    }
    if ((СерийныйНомер == null)) {
        this.Adapter.InsertCommand.Parameters[1].Value = global::System.DBNull.Value;
    }
    else {
        this.Adapter.InsertCommand.Parameters[1].Value = ((string)(СерийныйНомер));
    }
    if ((НаСкладе.HasValue == true)) {
        this.Adapter.InsertCommand.Parameters[2].Value = ((int)(НаСкладе.Value));
    }
    else {
        this.Adapter.InsertCommand.Parameters[2].Value = global::System.DBNull.Value;
    }
    if ((Цена.HasValue == true)) {
        this.Adapter.InsertCommand.Parameters[3].Value = ((decimal)(Цена.Value));
    }
    else {
        this.Adapter.InsertCommand.Parameters[3].Value = global::System.DBNull.Value;
    }
    if ((Заказано.HasValue == true)) {
        this.Adapter.InsertCommand.Parameters[4].Value = ((int)(Заказано.Value));
    }
}

```

```

else {
    this.Adapter.InsertCommand.Parameters[4].Value = global::System.DBNull.Value;
}
if ((МинимальныйЗапас.HasValue == true)) {
    this.Adapter.InsertCommand.Parameters[5].Value =
((int)(МинимальныйЗапас.Value));
}
else {
    this.Adapter.InsertCommand.Parameters[5].Value = global::System.DBNull.Value;
}
if ((ВремяЗадержки == null)) {
    this.Adapter.InsertCommand.Parameters[6].Value = global::System.DBNull.Value;
}
else {
    this.Adapter.InsertCommand.Parameters[6].Value = ((string)(ВремяЗадержки));
}
if ((КодТипа.HasValue == true)) {
    this.Adapter.InsertCommand.Parameters[7].Value = ((int)(КодТипа.Value));
}
else {
    this.Adapter.InsertCommand.Parameters[7].Value = global::System.DBNull.Value;
}

global::System.Data.ConnectionState previousConnectionState =
this.Adapter.InsertCommand.Connection.State;

if (((this.Adapter.InsertCommand.Connection.State &
global::System.Data.ConnectionState.Open)
    != global::System.Data.ConnectionState.Open)) {
    this.Adapter.InsertCommand.Connection.Open();
}

```

```

try {
    int returnValue = this.Adapter.InsertCommand.ExecuteNonQuery();
    return returnValue;
}
finally {
    if ((previousConnectionState == global::System.Data.ConnectionState.Closed)) {
        this.Adapter.InsertCommand.Connection.Close();
    }
}
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]
```

```
[global::System.ComponentModel.DataObjectMethodAttribute(global::System.ComponentModel.DataObjectMethodType.Update, true)]
```

```

public virtual int Update(
    string Марка,
    string СерийныйНомер,
    global::System.Nullable<int> НаСкладе,
    global::System.Nullable<decimal> Цена,
    global::System.Nullable<int> Заказано,
    global::System.Nullable<int> МинимальныйЗапас,
    string ВремяЗадержки,
    global::System.Nullable<int> КодТипа,
    int Original_КодТовара,
    string Original_Марка,

```

```

string Original_СерийныйНомер,
global::System.Nullable<int> Original_НаСкладе,
global::System.Nullable<decimal> Original_Цена,
global::System.Nullable<int> Original_Заказано,
global::System.Nullable<int> Original_МинимальныйЗапас,
string Original_ВремяЗадержки,
global::System.Nullable<int> Original_КодТипа) {
if ((Марка == null)) {
    this.Adapter.UpdateCommand.Parameters[0].Value = global::System.DBNull.Value;
}
else {
    this.Adapter.UpdateCommand.Parameters[0].Value = ((string)(Марка));
}
if ((СерийныйНомер == null)) {
    this.Adapter.UpdateCommand.Parameters[1].Value = global::System.DBNull.Value;
}
else {
    this.Adapter.UpdateCommand.Parameters[1].Value = ((string)(СерийныйНомер));
}
if ((НаСкладе.HasValue == true)) {
    this.Adapter.UpdateCommand.Parameters[2].Value = ((int)(НаСкладе.Value));
}
else {
    this.Adapter.UpdateCommand.Parameters[2].Value = global::System.DBNull.Value;
}
if ((Цена.HasValue == true)) {
    this.Adapter.UpdateCommand.Parameters[3].Value = ((decimal)(Цена.Value));
}
}

```

```

else {
    this.Adapter.UpdateCommand.Parameters[3].Value = global::System.DBNull.Value;
}
if ((Заказано.HasValue == true)) {
    this.Adapter.UpdateCommand.Parameters[4].Value = ((int)(Заказано.Value));
}
else {
    this.Adapter.UpdateCommand.Parameters[4].Value = global::System.DBNull.Value;
}
if ((МинимальныйЗапас.HasValue == true)) {
    this.Adapter.UpdateCommand.Parameters[5].Value =
((int)(МинимальныйЗапас.Value));
}
else {
    this.Adapter.UpdateCommand.Parameters[5].Value = global::System.DBNull.Value;
}
if ((ВремяЗадержки == null)) {
    this.Adapter.UpdateCommand.Parameters[6].Value = global::System.DBNull.Value;
}
else {
    this.Adapter.UpdateCommand.Parameters[6].Value = ((string)(ВремяЗадержки));
}
if ((КодТипа.HasValue == true)) {
    this.Adapter.UpdateCommand.Parameters[7].Value = ((int)(КодТипа.Value));
}
else {
    this.Adapter.UpdateCommand.Parameters[7].Value = global::System.DBNull.Value;
}
this.Adapter.UpdateCommand.Parameters[8].Value = ((int)(Original_КодТовара));

```

```

if ((Original_Марка == null)) {
    this.Adapter.UpdateCommand.Parameters[9].Value = ((object)(1));
    this.Adapter.UpdateCommand.Parameters[10].Value = global::System.DBNull.Value;
}
else {
    this.Adapter.UpdateCommand.Parameters[9].Value = ((object)(0));
    this.Adapter.UpdateCommand.Parameters[10].Value = ((string)(Original_Марка));
}
if ((Original_СерийныйНомер == null)) {
    this.Adapter.UpdateCommand.Parameters[11].Value = ((object)(1));
    this.Adapter.UpdateCommand.Parameters[12].Value = global::System.DBNull.Value;
}
else {
    this.Adapter.UpdateCommand.Parameters[11].Value = ((object)(0));
    this.Adapter.UpdateCommand.Parameters[12].Value =
((string)(Original_СерийныйНомер));
}
if ((Original_НаСкладе.HasValue == true)) {
    this.Adapter.UpdateCommand.Parameters[13].Value = ((object)(0));
    this.Adapter.UpdateCommand.Parameters[14].Value =
((int)(Original_НаСкладе.Value));
}
else {
    this.Adapter.UpdateCommand.Parameters[13].Value = ((object)(1));
    this.Adapter.UpdateCommand.Parameters[14].Value = global::System.DBNull.Value;
}
if ((Original_Цена.HasValue == true)) {
    this.Adapter.UpdateCommand.Parameters[15].Value = ((object)(0));

```

```

        this.Adapter.UpdateCommand.Parameters[16].Value =
((decimal)(Original_Цена.Value));
    }
    else {
        this.Adapter.UpdateCommand.Parameters[15].Value = ((object)(1));
        this.Adapter.UpdateCommand.Parameters[16].Value = global::System.DBNull.Value;
    }
    if ((Original_Заказано.HasValue == true)) {
        this.Adapter.UpdateCommand.Parameters[17].Value = ((object)(0));
        this.Adapter.UpdateCommand.Parameters[18].Value =
((int)(Original_Заказано.Value));
    }
    else {
        this.Adapter.UpdateCommand.Parameters[17].Value = ((object)(1));
        this.Adapter.UpdateCommand.Parameters[18].Value = global::System.DBNull.Value;
    }
    if ((Original_МинимальныйЗапас.HasValue == true)) {
        this.Adapter.UpdateCommand.Parameters[19].Value = ((object)(0));
        this.Adapter.UpdateCommand.Parameters[20].Value =
((int)(Original_МинимальныйЗапас.Value));
    }
    else {
        this.Adapter.UpdateCommand.Parameters[19].Value = ((object)(1));
        this.Adapter.UpdateCommand.Parameters[20].Value = global::System.DBNull.Value;
    }
    if ((Original_ВремяЗадержки == null)) {
        this.Adapter.UpdateCommand.Parameters[21].Value = ((object)(1));
        this.Adapter.UpdateCommand.Parameters[22].Value = global::System.DBNull.Value;
    }

```

```

else {
    this.Adapter.UpdateCommand.Parameters[21].Value = ((object)(0));
    this.Adapter.UpdateCommand.Parameters[22].Value =
((string)(Original_ВремяЗадержки));
}
if ((Original_КодТипа.HasValue == true)) {
    this.Adapter.UpdateCommand.Parameters[23].Value = ((object)(0));
    this.Adapter.UpdateCommand.Parameters[24].Value =
((int)(Original_КодТипа.Value));
}
else {
    this.Adapter.UpdateCommand.Parameters[23].Value = ((object)(1));
    this.Adapter.UpdateCommand.Parameters[24].Value = global::System.DBNull.Value;
}
global::System.Data.ConnectionState previousConnectionState =
this.Adapter.UpdateCommand.Connection.State;
if (((this.Adapter.UpdateCommand.Connection.State &
global::System.Data.ConnectionState.Open)
    != global::System.Data.ConnectionState.Open)) {
    this.Adapter.UpdateCommand.Connection.Open();
}
try {
    int returnValue = this.Adapter.UpdateCommand.ExecuteNonQuery();
    return returnValue;
}
finally {
    if ((previousConnectionState == global::System.Data.ConnectionState.Closed)) {
        this.Adapter.UpdateCommand.Connection.Close();
    }
}
}

```

```

    }
}

/// <summary>
    ///TableAdapterManager is used to coordinate TableAdapters in the dataset to enable
    Hierarchical Update scenarios
    ///</summary>

    [global::System.ComponentModel.DesignerCategoryAttribute("code")]
    [global::System.ComponentModel.ToolboxItem(true)]

    [global::System.ComponentModel.DesignerAttribute("Microsoft.VSDesigner.DataSource.Design.
    TableAdapterManagerDesigner, Microsoft.VSD" +
        "esigner, Version=10.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")]

    [global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapterManager")]

    public partial class TableAdapterManager : global::System.ComponentModel.Component {

        private UpdateOrderOption _updateOrder;

        private ЗапчастиTableAdapter _запчастиTableAdapter;

        private bool _backupDataSetBeforeUpdate;

        private global::System.Data.IDbConnection _connection;

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

        public UpdateOrderOption UpdateOrder {

```

```

get {
    return this._updateOrder;
}
set {
    this._updateOrder = value;
}
}

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

[global::System.ComponentModel.EditorAttribute("Microsoft.VSDesigner.DataSource.Design.TableAdapterManagerPropertyEditor, Microso" +

"ft.VSDesigner, Version=10.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3"

+

"a", "System.Drawing.Design.UITypeEditor")]

public ЗапчастиTableAdapter ЗапчастиTableAdapter {

get {

return this._запчастиTableAdapter;

}

set {

this._запчастиTableAdapter = value;

}

}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

```

public bool BackupDataSetBeforeUpdate {
    get {
        return this._backupDataSetBeforeUpdate;
    }
    set {
        this._backupDataSetBeforeUpdate = value;
    }
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
[global::System.ComponentModel.Browsable(false)]
```

```
public global::System.Data.IDbConnection Connection {
```

```
    get {
```

```
        if ((this._connection != null)) {
```

```
            return this._connection;
```

```
        }
```

```
        if (((this._запчастиTableAdapter != null)
```

```
            && (this._запчастиTableAdapter.Connection != null))) {
```

```
            return this._запчастиTableAdapter.Connection;
```

```
        }
```

```
        return null;
```

```
    }
```

```
    set {
```

```
        this._connection = value;
```

```
    }
```

```
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
[global::System.ComponentModel.Browsable(false)]
```

```
public int TableAdapterInstanceCount {
```

```
    get {
```

```
        int count = 0;
```

```
        if ((this._запчастиTableAdapter != null)) {
```

```
            count = (count + 1);
```

```
        }
```

```
        return count;
```

```
    }
```

```
}
```

```
/// <summary>
```

```
///Update rows in top-down order.
```

```
///</summary>
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
    private int UpdateUpdatedRows(MySteelDataDataSet dataSet,  
global::System.Collections.Generic.List<global::System.Data.DataRow> allChangedRows,  
global::System.Collections.Generic.List<global::System.Data.DataRow> allAddedRows) {
```

```
        int result = 0;
```

```
        if ((this._запчастиTableAdapter != null)) {
```

```
            global::System.Data.DataRow[] updatedRows = dataSet.Запчасти.Select(null, null,  
global::System.Data.DataViewRowState.ModifiedCurrent);
```

```
            updatedRows = this.GetRealUpdatedRows(updatedRows, allAddedRows);
```

```

    if (((updatedRows != null)
        && (0 < updatedRows.Length))) {
        result = (result + this._запчастиTableAdapter.Update(updatedRows));
        allChangedRows.AddRange(updatedRows);
    }
}
return result;
}

```

```

/// <summary>

```

```

///Insert rows in top-down order.

```

```

///</summary>

```

```

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

```

```

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]

```

```

    private int UpdateInsertedRows(MySteelDataDataSet dataSet,
global::System.Collections.Generic.List<global::System.Data.DataRow> allAddedRows) {
        int result = 0;
        if ((this._запчастиTableAdapter != null)) {
            global::System.Data.DataRow[] addedRows = dataSet.Запчасти.Select(null, null,
global::System.Data.DataViewRowState.Added);
            if (((addedRows != null)
                && (0 < addedRows.Length))) {
                result = (result + this._запчастиTableAdapter.Update(addedRows));
                allAddedRows.AddRange(addedRows);
            }
        }
        return result;
    }
}

```

```

/// <summary>

///Delete rows in bottom-up order.

///</summary>

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataS
etGenerator", "15.0.0.0")]

private int UpdateDeletedRows(MySteelDataDataSet dataSet,
global::System.Collections.Generic.List<global::System.Data.DataRow> allChangedRows) {

    int result = 0;

    if ((this._запчастиTableAdapter != null)) {

        global::System.Data.DataRow[] deletedRows = dataSet.Запчасти.Select(null, null,
global::System.Data.DataViewRowState.Deleted);

        if (((deletedRows != null)

            && (0 < deletedRows.Length))) {

            result = (result + this._запчастиTableAdapter.Update(deletedRows));

            allChangedRows.AddRange(deletedRows);

        }

    }

    return result;

}

/// <summary>

///Remove inserted rows that become updated rows after calling
TableAdapter.Update(inserted rows) first

///</summary>

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataS
etGenerator", "15.0.0.0")]

```

```

private global::System.Data.DataRow[]
GetRealUpdatedRows(global::System.Data.DataRow[] updatedRows,
global::System.Collections.Generic.List<global::System.Data.DataRow> allAddedRows) {
    if (((updatedRows == null)
        || (updatedRows.Length < 1))) {
        return updatedRows;
    }
    if (((allAddedRows == null)
        || (allAddedRows.Count < 1))) {
        return updatedRows;
    }
    global::System.Collections.Generic.List<global::System.Data.DataRow>
realUpdatedRows = new
global::System.Collections.Generic.List<global::System.Data.DataRow>();
    for (int i = 0; (i < updatedRows.Length); i = (i + 1)) {
        global::System.Data.DataRow row = updatedRows[i];
        if ((allAddedRows.Contains(row) == false)) {
            realUpdatedRows.Add(row);
        }
    }
    return realUpdatedRows.ToArray();
}

/// <summary>
///Update all changes to the dataset.
///</summary>

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataS
etGenerator", "15.0.0.0")]

```

```

public virtual int UpdateAll(MySteelDataDataSet dataSet) {
    if ((dataSet == null)) {
        throw new global::System.ArgumentNullException("dataSet");
    }
    if ((dataSet.HasChanges() == false)) {
        return 0;
    }
    if (((this._запчастиTableAdapter != null)
        &&
        (this.MatchTableAdapterConnection(this._запчастиTableAdapter.Connection) == false))) {
        throw new global::System.ArgumentException("Все адаптеры таблицы,
управляемые диспетчером адаптера таблицы TableAdapterManager" +
            "г, должны использовать одинаковую строку подключения.");
    }
    global::System.Data.IDbConnection workConnection = this.Connection;
    if ((workConnection == null)) {
        throw new global::System.ApplicationException("TableAdapterManager не содержит
сведений о подключении. Укажите для каждого адапт" +
            "ера таблицы TableAdapterManager допустимый экземпляр адаптера
таблицы.");
    }
    bool workConnOpened = false;
    if (((workConnection.State & global::System.Data.ConnectionState.Broken)
        == global::System.Data.ConnectionState.Broken)) {
        workConnection.Close();
    }
    if ((workConnection.State == global::System.Data.ConnectionState.Closed)) {
        workConnection.Open();
        workConnOpened = true;
    }
}

```

```

        global::System.Data.IDbTransaction workTransaction =
workConnection.BeginTransaction();

        if ((workTransaction == null)) {

            throw new global::System.ApplicationException("Не удается начать транзакцию.
Текущее соединение данных не поддерживает транзакции" +

                "и или текущее состояние не позволяет начать транзакцию.");

        }

        global::System.Collections.Generic.List<global::System.Data.DataRow>
allChangedRows = new
global::System.Collections.Generic.List<global::System.Data.DataRow>();

        global::System.Collections.Generic.List<global::System.Data.DataRow> allAddedRows
= new global::System.Collections.Generic.List<global::System.Data.DataRow>();

        global::System.Collections.Generic.List<global::System.Data.Common.DataAdapter>
adaptersWithAcceptChangesDuringUpdate = new
global::System.Collections.Generic.List<global::System.Data.Common.DataAdapter>();

        global::System.Collections.Generic.Dictionary<object,
global::System.Data.IDbConnection> revertConnections = new
global::System.Collections.Generic.Dictionary<object, global::System.Data.IDbConnection>();

        int result = 0;

        global::System.Data.DataSet backupDataSet = null;

        if (this.BackupDataSetBeforeUpdate) {

            backupDataSet = new global::System.Data.DataSet();

            backupDataSet.Merge(dataSet);

        }

        try {

            // ---- Prepare for update -----

            //

            if ((this._запчастиTableAdapter != null)) {

                revertConnections.Add(this._запчастиTableAdapter,
this._запчастиTableAdapter.Connection);

                this._запчастиTableAdapter.Connection =
((global::System.Data.OleDb.OleDbConnection)(workConnection));

```

```

        this._запчастиTableAdapter.Transaction =
((global::System.Data.OleDb.OleDbTransaction)(workTransaction));

        if (this._запчастиTableAdapter.Adapter.AcceptChangesDuringUpdate) {

            this._запчастиTableAdapter.Adapter.AcceptChangesDuringUpdate = false;

adaptersWithAcceptChangesDuringUpdate.Add(this._запчастиTableAdapter.Adapter);

        }

    }

    //

    //---- Perform updates -----

    //

    if ((this.UpdateOrder == UpdateOrderOption.UpdateInsertDelete)) {

        result = (result + this.UpdateUpdatedRows(dataSet, allChangedRows,
allAddedRows));

        result = (result + this.UpdateInsertedRows(dataSet, allAddedRows));

    }

    else {

        result = (result + this.UpdateInsertedRows(dataSet, allAddedRows));

        result = (result + this.UpdateUpdatedRows(dataSet, allChangedRows,
allAddedRows));

    }

    result = (result + this.UpdateDeletedRows(dataSet, allChangedRows));

    //

    //---- Commit updates -----

    //

    workTransaction.Commit();

    if ((0 < allAddedRows.Count)) {

        global::System.Data.DataRow[] rows = new
System.Data.DataRow[allAddedRows.Count];

        allAddedRows.CopyTo(rows);

```

```

    for (int i = 0; (i < rows.Length); i = (i + 1)) {
        global::System.Data.DataRow row = rows[i];
        row.AcceptChanges();
    }
}

if ((0 < allChangedRows.Count)) {
    global::System.Data.DataRow[] rows = new
System.Data.DataRow[allChangedRows.Count];

    allChangedRows.CopyTo(rows);

    for (int i = 0; (i < rows.Length); i = (i + 1)) {
        global::System.Data.DataRow row = rows[i];
        row.AcceptChanges();
    }
}

}

catch (global::System.Exception ex) {
    workTransaction.Rollback();

    // ---- Restore the dataset -----

    if (this.BackupDataSetBeforeUpdate) {
        global::System.Diagnostics.Debug.Assert((backupDataSet != null));
        dataSet.Clear();
        dataSet.Merge(backupDataSet);
    }
    else {
        if ((0 < allAddedRows.Count)) {
            global::System.Data.DataRow[] rows = new
System.Data.DataRow[allAddedRows.Count];

            allAddedRows.CopyTo(rows);

            for (int i = 0; (i < rows.Length); i = (i + 1)) {

```

```

        global::System.Data.DataRow row = rows[i];

        row.AcceptChanges();

        row.SetAdded();

    }

}

}

throw ex;

}

finally {

    if (workConnOpened) {

        workConnection.Close();

    }

    if ((this._запчастиTableAdapter != null)) {

        this._запчастиTableAdapter.Connection =

        ((global::System.Data.OleDb.OleDbConnection)(revertConnections[this._запчастиTableAdapte

r]));

        this._запчастиTableAdapter.Transaction = null;

    }

    if ((0 < adaptersWithAcceptChangesDuringUpdate.Count)) {

        global::System.Data.Common.DataAdapter[] adapters = new

System.Data.Common.DataAdapter[adaptersWithAcceptChangesDuringUpdate.Count];

        adaptersWithAcceptChangesDuringUpdate.CopyTo(adapters);

        for (int i = 0; (i < adapters.Length); i = (i + 1)) {

            global::System.Data.Common.DataAdapter adapter = adapters[i];

            adapter.AcceptChangesDuringUpdate = true;

        }

    }

}

return result;

```

```
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
protected virtual void SortSelfReferenceRows(global::System.Data.DataRow[] rows,  
global::System.Data.DataRelation relation, bool childFirst) {
```

```
    global::System.Array.Sort<global::System.Data.DataRow>(rows, new  
    SelfReferenceComparer(relation, childFirst));
```

```
}
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
protected virtual bool MatchTableAdapterConnection(global::System.Data.IDbConnection  
inputConnection) {
```

```
    if ((this._connection != null)) {
```

```
        return true;
```

```
    }
```

```
    if (((this.Connection == null)
```

```
        || (inputConnection == null))) {
```

```
        return true;
```

```
    }
```

```
    if (string.Equals(this.Connection.ConnectionString, inputConnection.ConnectionString,  
global::System.StringComparison.Ordinal)) {
```

```
        return true;
```

```
    }
```

```
    return false;
```

```
}
```

```
/// <summary>
```

```
///Update Order Option
```

```
///</summary>
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public enum UpdateOrderOption {
```

```
    InsertUpdateDelete = 0,
```

```
    UpdateInsertDelete = 1,
```

```
}
```

```
/// <summary>
```

```
///Used to sort self-referenced table's rows
```

```
///</summary>
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
private class SelfReferenceComparer : object,  
global::System.Collections.Generic.IComparer<global::System.Data.DataRow> {
```

```
    private global::System.Data.DataRelation _relation;
```

```
    private int _childFirst;
```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```

        internal SelfReferenceComparer(global::System.Data.DataRelation relation, bool
childFirst) {
            this._relation = relation;
            if (childFirst) {
                this._childFirst = -1;
            }
            else {
                this._childFirst = 1;
            }
        }
    }
}

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataS
etGenerator", "15.0.0.0")]
```

```

        private global::System.Data.DataRow GetRoot(global::System.Data.DataRow row, out
int distance) {

```

```
            global::System.Diagnostics.Debug.Assert((row != null));
```

```
            global::System.Data.DataRow root = row;
```

```
            distance = 0;
```

```

            global::System.Collections.Generic.IDictionary<global::System.Data.DataRow,
global::System.Data.DataRow> traversedRows = new
global::System.Collections.Generic.Dictionary<global::System.Data.DataRow,
global::System.Data.DataRow>();

```

```
            traversedRows[row] = row;
```

```

            global::System.Data.DataRow parent = row.GetParentRow(this._relation,
global::System.Data.DataRowVersion.Default);

```

```
            for (
```

```
                ; ((parent != null)
```

```

        && (traversedRows.ContainsKey(parent) == false));
    ) {
        distance = (distance + 1);
        root = parent;
        traversedRows[parent] = parent;
        parent = parent.GetParentRow(this._relation,
global::System.Data.DataRowVersion.Default);
    }

    if ((distance == 0)) {
        traversedRows.Clear();
        traversedRows[row] = row;
        parent = row.GetParentRow(this._relation,
global::System.Data.DataRowVersion.Original);
        for (
            ; ((parent != null)
                && (traversedRows.ContainsKey(parent) == false));
            ) {
                distance = (distance + 1);
                root = parent;
                traversedRows[parent] = parent;
                parent = parent.GetParentRow(this._relation,
global::System.Data.DataRowVersion.Original);
            }
        }

        return root;
    }

```

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
```

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "15.0.0.0")]
```

```
public int Compare(global::System.Data.DataRow row1, global::System.Data.DataRow row2) {
```

```
    if (object.ReferenceEquals(row1, row2)) {
```

```
        return 0;
```

```
    }
```

```
    if ((row1 == null)) {
```

```
        return -1;
```

```
    }
```

```
    if ((row2 == null)) {
```

```
        return 1;
```

```
    }
```

```
    int distance1 = 0;
```

```
    global::System.Data.DataRow root1 = this.GetRoot(row1, out distance1);
```

```
    int distance2 = 0;
```

```
    global::System.Data.DataRow root2 = this.GetRoot(row2, out distance2);
```

```
    if (object.ReferenceEquals(root1, root2)) {
```

```
        return (this._childFirst * distance1.CompareTo(distance2));
```

```
    }
```

```
    else {
```

```
        global::System.Diagnostics.Debug.Assert(((root1.Table != null)
```

```
            && (root2.Table != null)));
```

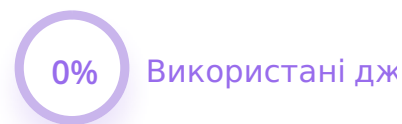
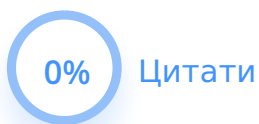
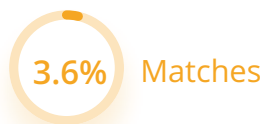
```
        if ((root1.Table.Rows.IndexOf(root1) < root2.Table.Rows.IndexOf(root2))) {
```

```
        return -1;
    }
    else {
        return 1;
    }
}
}
}
}
```

4.5 Пояснювальна записка_перев

Завантажено: 06/23/2020 | Перевірено: 06/23/2020

● Matches ● Цитата ● Використані джерела ● Заміна символів



Matches

Веб джерела

46

1	ukrbukva.net https://ukrbukva.net/page,5,92030-Sredstva-postroeniya-sistem-elektronnoy-kommercii.html	1.31%
2	cad.kpi.ua http://cad.kpi.ua/attachments/093_2017d_Chantsova.pdf	1.25%
3	inmad.vntu.edu.ua http://inmad.vntu.edu.ua/portal/static/EE71D07C-5FE7-4F40-8FC9-8F1D3CC93FAC.pdf	1.12%
4	www.turkaramamotoru.com https://www.turkaramamotoru.com/uk/%D0%91%D0%B0%D0%B3%D0%B0%D1%82%D0%BE%D0%BA%D1%80%D0%B8%...	1%
5	uk.wikipedia.org https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B3%D0%B0%D1%82%D0%BE%D0%BA%D1%80%D0%B8%D1%82%...	0.93%
6	uk.m.wikipedia.org https://uk.m.wikipedia.org/wiki/%D0%91%D0%B0%D0%B3%D0%B0%D1%82%D0%BE%D0%BA%D1%80%D0%B8%D1%8...	0.93%
7	dir.md https://dir.md/wiki/%D0%91%D0%B0%D0%B3%D0%B0%D1%82%D0%BE%D0%BA%D1%80%D0%B8%D1%82%D0%B5%...	0.91%
8	ela.kpi.ua https://ela.kpi.ua/bitstream/123456789/23556/1/Dushutin_magistr.pdf	0.73%
9	www.wikizero.com https://www.wikizero.com/uk/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B8_%D1%80%D0%BE%D0%B7%D0%B...	0.52%
10	uk.wikipedia.org https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B8_%D1%80%D0%BE%D0%B7%D0%B2...	0.52%

4.5 Пояснювальна записка_перев

Завантажено: 06/23/2020 | Перевірено: 06/23/2020

● Matches ● Цитата ● Використані джерела ● Заміна символів

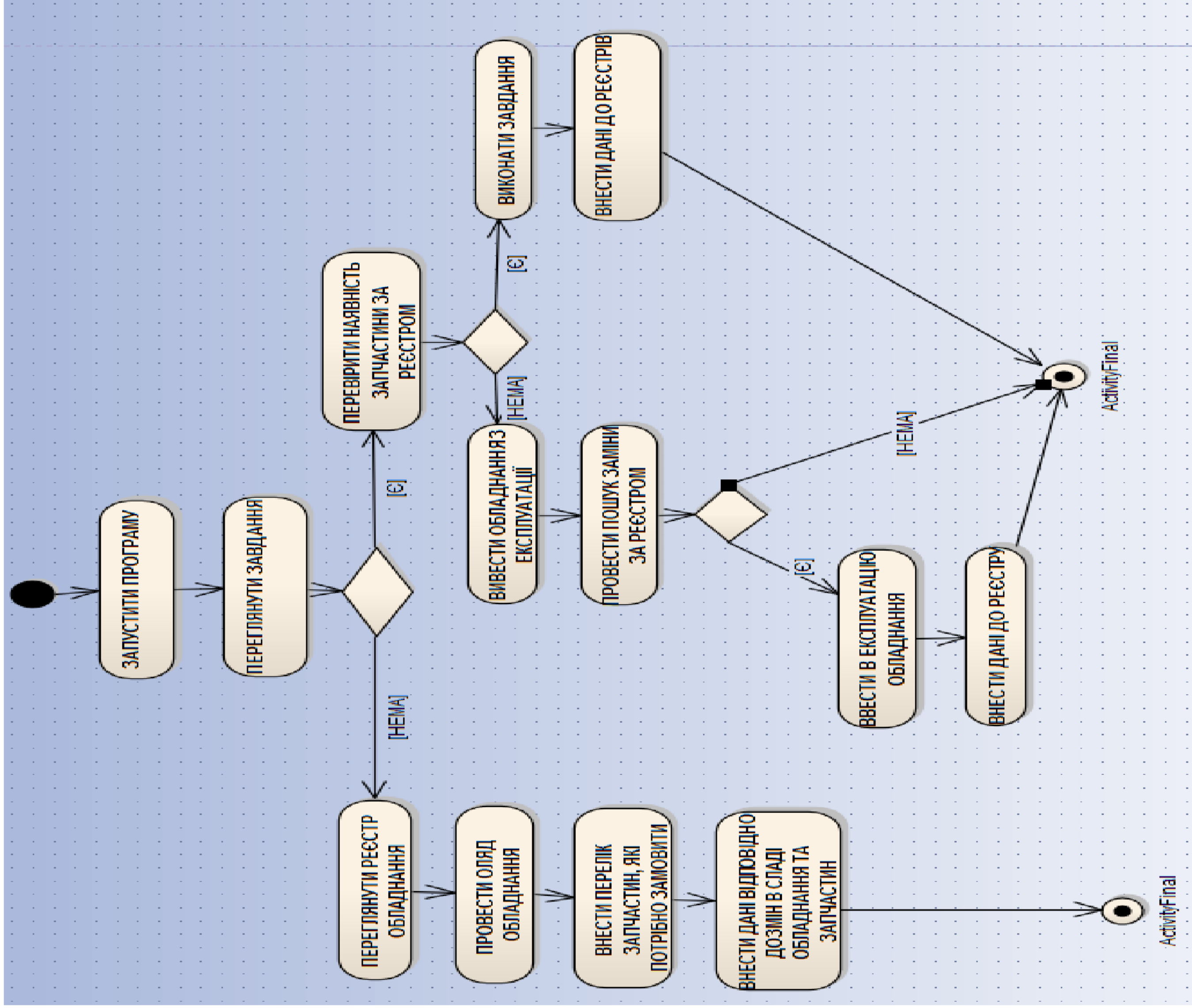
11	studfile.net https://studfile.net/preview/5009975/page:8/	0.5%
12	www.researchgate.net https://www.researchgate.net/publication/328967405_METODI_BAGATOKRITERIALNOI_OPTIMIZACII	0.5%
13	mylektsii.ru https://mylektsii.ru/8-10976.html	0.5%
14	b-ok.org https://b-ok.org/book/2997050/945d3a	0.5%
15	ela.kpi.ua https://ela.kpi.ua/bitstream/123456789/29208/1/Tataurov_bakalavr.pdf	0.21%
16	ua-referat.com https://ua-referat.com/%D0%A0%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B0_%D0%B5%D1%84%D0%...	0.12%
17	ua.kursoviks.com.ua https://ua.kursoviks.com.ua/metodychni_vkazivky/article_post/3544-sistema-keruvannya-bazami-danikh-mysql-proyektiv...	0.12%
18	i1.poltava.to https://i1.poltava.to/uploads/2019/10/2019-10-10/tz-smart-region.pdf	0.12%
19	sportlandprim.weebly.com http://sportlandprim.weebly.com/blog	0.1%
20	www.znannya.org http://www.znannya.org/?view=mysql	0.1%
21	www.znannya.org http://www.znannya.org/?view=mysql-intro	0.1%
22	pmit.fmi.org.ua http://pmit.fmi.org.ua/files/5abb7a0010e5a3.36041163.doc	0.1%
23	lib.mdpu.org.ua http://lib.mdpu.org.ua/e-book/vstup/L5.htm	0.1%
24	inmad.vntu.edu.ua http://inmad.vntu.edu.ua/portal/static/E25D1BCD-7DBB-4BBB-95B3-1C0CB446B5C9.pdf	0.1%
25	docplayer.net https://docplayer.net/52334842-Stvorenniya-saytu-budivelnoyi-firmi.html	0.1%
26	eprints.library.odeku.edu.ua http://eprints.library.odeku.edu.ua/5974/1/Kirija_Rozrobka_web_orientovanoy_inform_sysem.pdf	0.1%
27	chanelparfum.weebly.com http://chanelparfum.weebly.com/blog/previous/4	0.1%
28	en.ppt-online.org https://en.ppt-online.org/447137	0.1%
29	megapredmet.ru https://megapredmet.ru/1-50082.html	0.1%

4.5 Пояснювальна записка_перев

Завантажено: 06/23/2020 | Перевірено: 06/23/2020

● Matches ● Цитата ● Використані джерела ● Заміна символів

30	knowledge.allbest.ru https://knowledge.allbest.ru/programming/2c0a65625b2bd69b5d43a89521206d37_0.html	0.1%
31	studopedya.ru https://studopedya.ru/2-39915.html	0.1%
32	www.sesiya.ru https://www.sesiya.ru/kurosovaya-rabota/Web-programmirovanie/stvorenniya-veb-forumu/	0.1%
33	www.bestreferat.ru https://www.bestreferat.ru/referat-261529.html	0.1%
34	www.bestreferat.ru https://www.bestreferat.ru/referat-142309.html	0.1%
35	vseosvita.ua https://vseosvita.ua/library/metodichna-rozrobka-provedenna-vidkritogo-practicnogo-zanatta-z-disciplini-obcisluvalna-tehnik...	0.1%
36	uk.wikipedia.org https://uk.wikipedia.org/wiki/MySQL	0.1%
37	ua.kurosviks.com.ua https://ua.kurosviks.com.ua/metodychni_vkazivky/article_post/1671-teksti-programnogo-kodu-kompleksu-zadach-pidtrimk...	0.1%
38	studopedia.com.ua https://studopedia.com.ua/1_127212_server-MYSQL.html	0.1%
39	knowledge.allbest.ru https://knowledge.allbest.ru/programming/2c0b65625b3bd68a4d43a88521306c36_0.html	0.1%
40	studfile.net https://studfile.net/preview/7011289/page:3/	0.1%
41	studfile.net https://studfile.net/preview/5607354/page:3/	0.1%
42	stud.wiki https://stud.wiki/programming/2c0b65625b3bd68a4d43a88521306c36_0.html	0.1%
43	revolution.allbest.ru https://revolution.allbest.ru/programming/00410703_0.html	0.1%
44	life-prog.ru https://life-prog.ru/2_43995_sistema-keruvannya-bazami-danih-MYSQL.html	0.1%
45	knowledge.allbest.ru https://knowledge.allbest.ru/programming/3c0b65625b3ac69a5c53a89421316d26_0.html	0.1%
46	arton.com.ua http://arton.com.ua/files/manuals/opis_movi.doc	0.1%



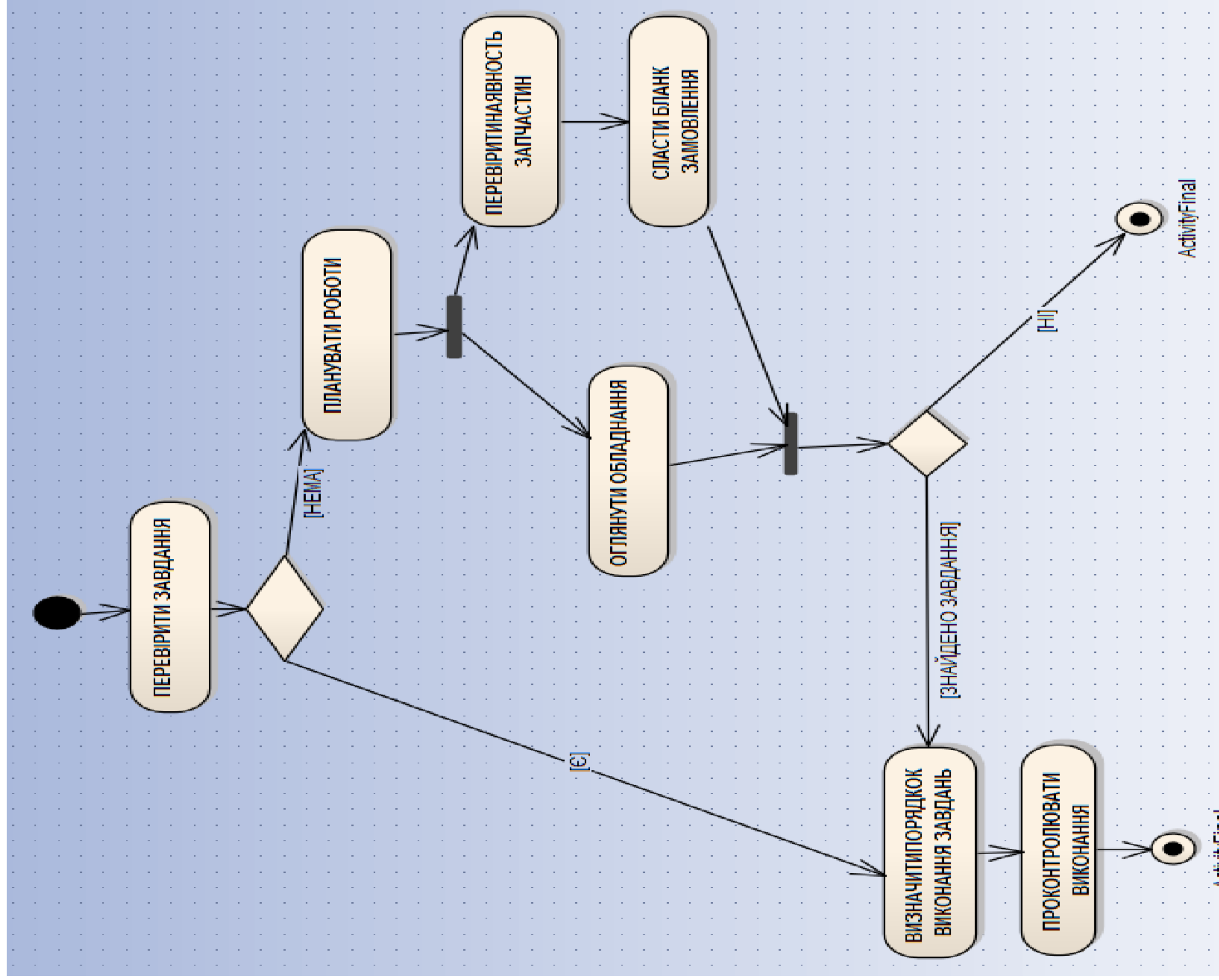
ДП 6106.02.002 ССД

Зм.	Арк.	№ докум.	Підп.	Дата
Розроб.		Оробей Е.В.		
Перев.		Новикова П.А.		
Т. Кон.				
Н. Кон.		Телищева Т.О.		
Затв.		Новикова П.А.		

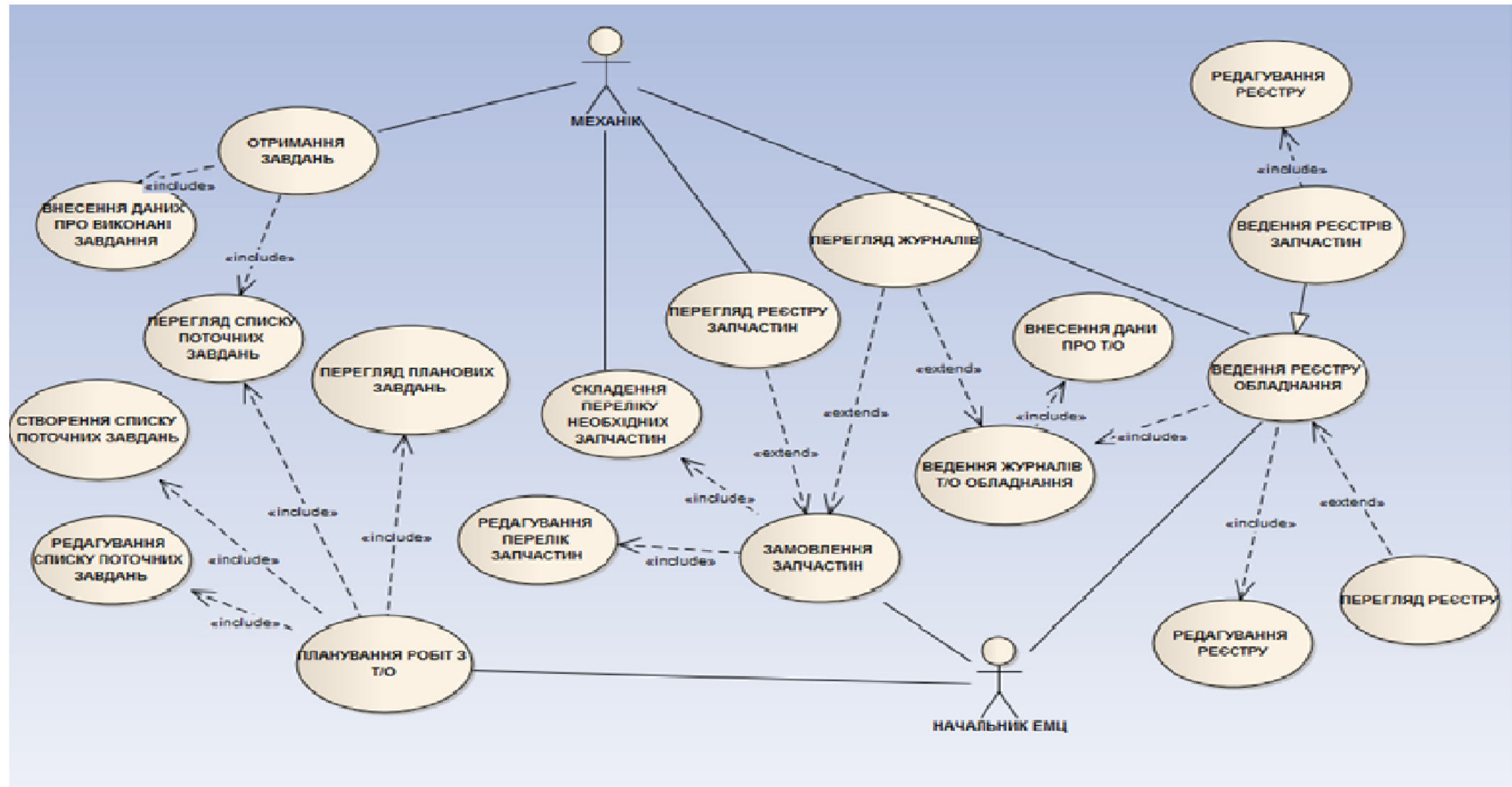
Лист.	Маса	Масштаб

Аркуш	Аркушів

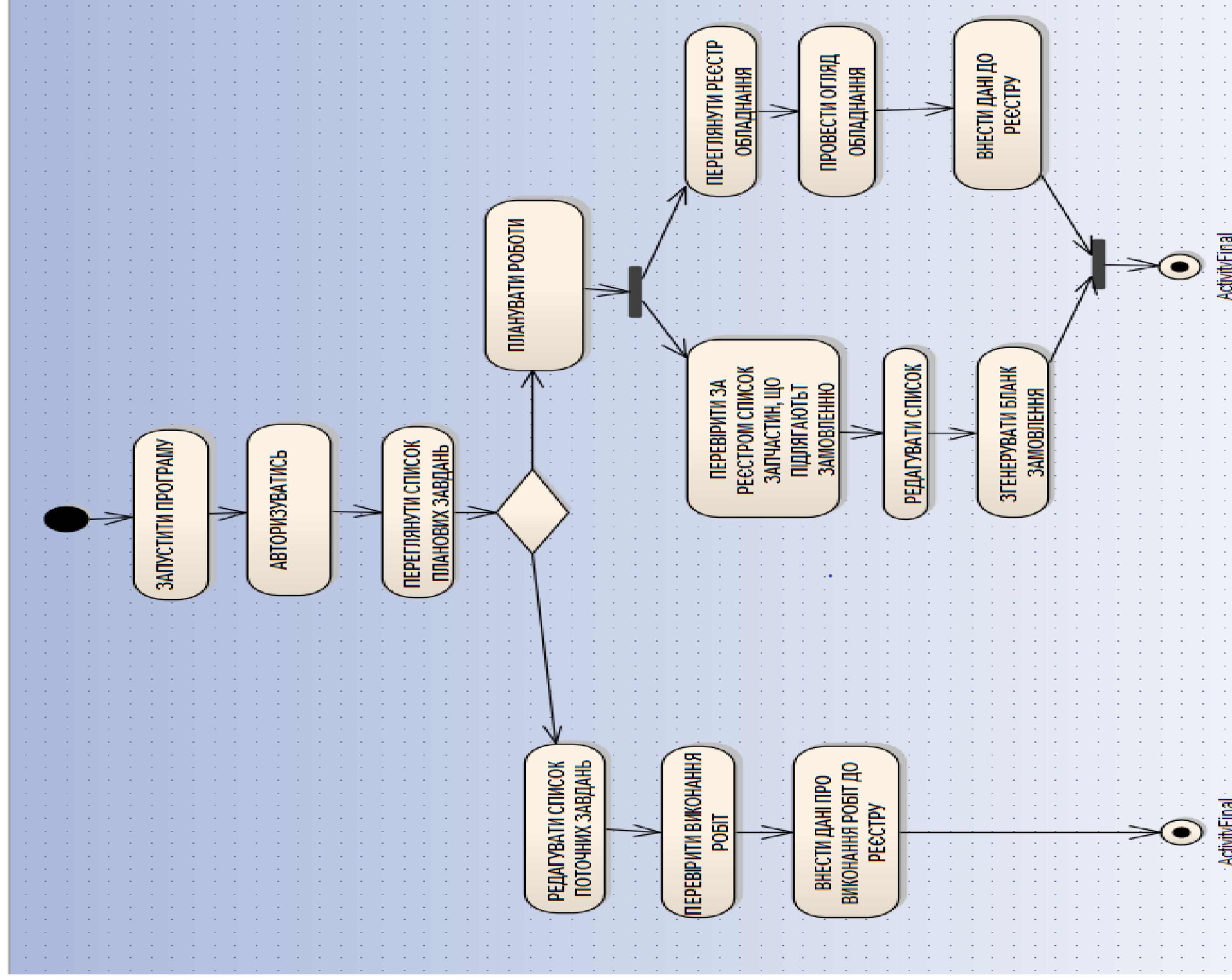
схема структурна діяльності	КПІ ім. Івора Сікорського Каф. АСОУ Гр. ІС-361
Автоматизована система управління технічним обслуговуванням виробничого обладнання	



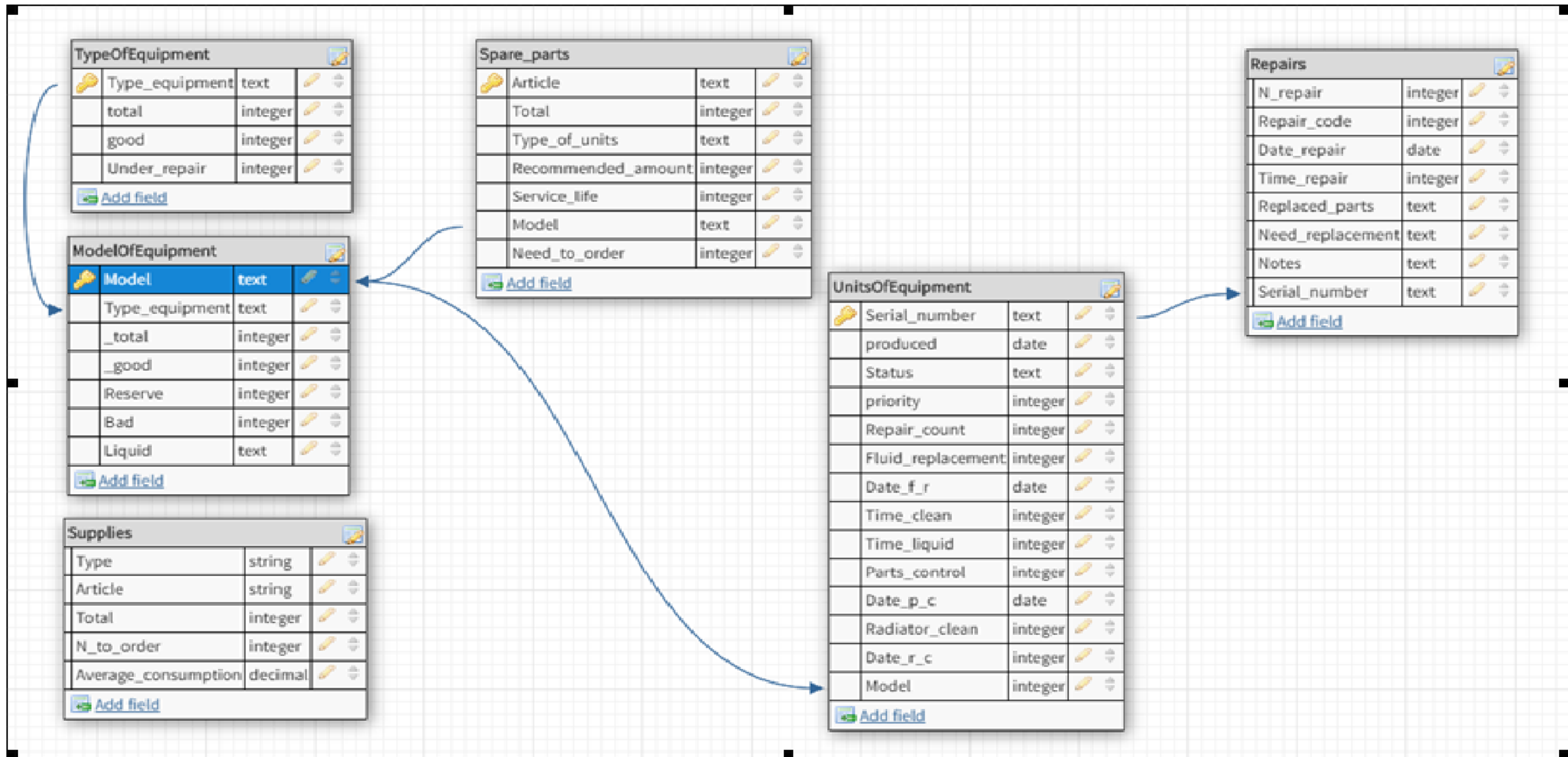
ДП 6106.02.001 ССД									
Зм.	Арк.	№ докум.	Підп.	Дата	Лист.	Маса	Масштаб	Аркуші	Аркушів
Розроб.		Оробей Е.В.							
Перев.		Новикова П.А.							
Т. Кон.									
Н. Кон.		Телишева Т.О.							КПІ ім. Ігоря Сікорського
Затв.		Новикова П.А.							Каф. АСОІУ
									Гр. ІС-361
									Автоматизована система управління технічним обслуговуванням виробничого обладнання



					ДП 6106.03.000 ССВ			
Зм.	Арк.	№ докум.	Підп.	Дата	схема структурна варіантів використання	Лист.	Маса	Масштаб
Розроб.		Оробей Е.В.						
Перев.		Новікова П.А.						
Т. Кон.						Аркуш	Аркушів	
Н. Кон.		Телишева Т.О.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361		
Затв.		Новікова П.А.						
					Автоматизована система управління технічним обслуговуванням виробничого обладнання			



ДП 6106.02.003 ССД				Лист.	Маса	Масштаб
схема структурна діяльності				Аркуш	Аркушів	
Автоматизована система управління технічним обслуговуванням виробничого обладнання				КПІ ім. Іоанн Сікорського Каф. АСОУ Гр. ІС-361		
Зм.	Арк.	№ докum.	Підп.	Дата		
Розроб.		Оробей Е.В.				
Перев.		Новикова П.А.				
Т. Кон.						
Н. Кон.		Телишева Т.О.				
Затв.		Новикова П.А.				



					ДП 6106.04.000 СБД			
					креслення вигляду екранних форм	Лит.	Маса	Масштаб
Зм.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Оробей Е.В.						
Перев.		Новікова П.А.						
Т. Кон.						Аркуш	Аркушів	
Н. Кон.		Телишева Т.О.			Автоматизована система управління технічним обслуговуванням виробничого обладнання	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361		
Затв.		Новікова П.А.						



Software interface with a green background and buttons: 'ВИХІД З ФОРМИ' (Exit Form), 'ДОДАТИ ОБЛАДНАННЯ' (Add Equipment), 'ДОДАТИ ЗАПЧАСТИНИ' (Add Spare Parts), 'ПЕРЕГЛЯНУТИ РЕЄСТР ОБЛАДНАННЯ' (View Equipment Register), 'ПЕРЕГЛЯНУТИ РЕЄСТР ЗАПЧАСТИН' (View Spare Parts Register), and 'ЗБЕРЕГТИ' (Save).

Код Товара	Марка	Серійний Номер	НаСкладі	Цена	Заказано	Минимальний Зап	Время Задержки	Код Типа
1	strobел	15253897	3	25				
2	strobел	451200158	2	256000	0	1	5мес	5
3	zic-zac	258035480	1	150000	1	1	12мес	3
6	fortuna	2522364	5	522	0	2	2	8
7	пшпн	46348	5					
8	пшпшгрен	54646	52					
9	пппн	9	45					
10	ро	78778	6					
11	ььжь	л	6467					
12	уубиу							
13			45					

Software interface with an orange background. Fields for data entry:

ТИП ОБЛАДНАННЯ:

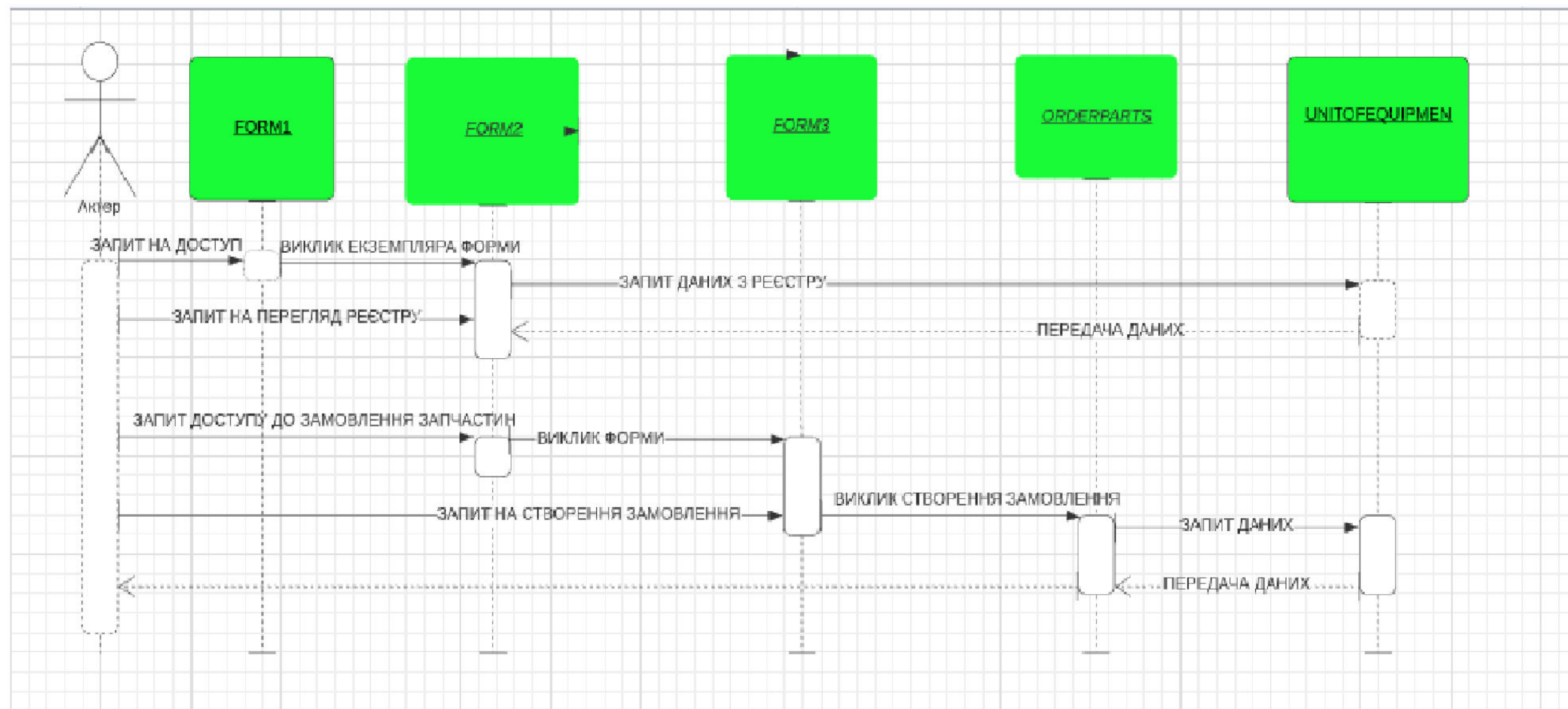
МОДЕЛЬ:

АРТИКУЛ:

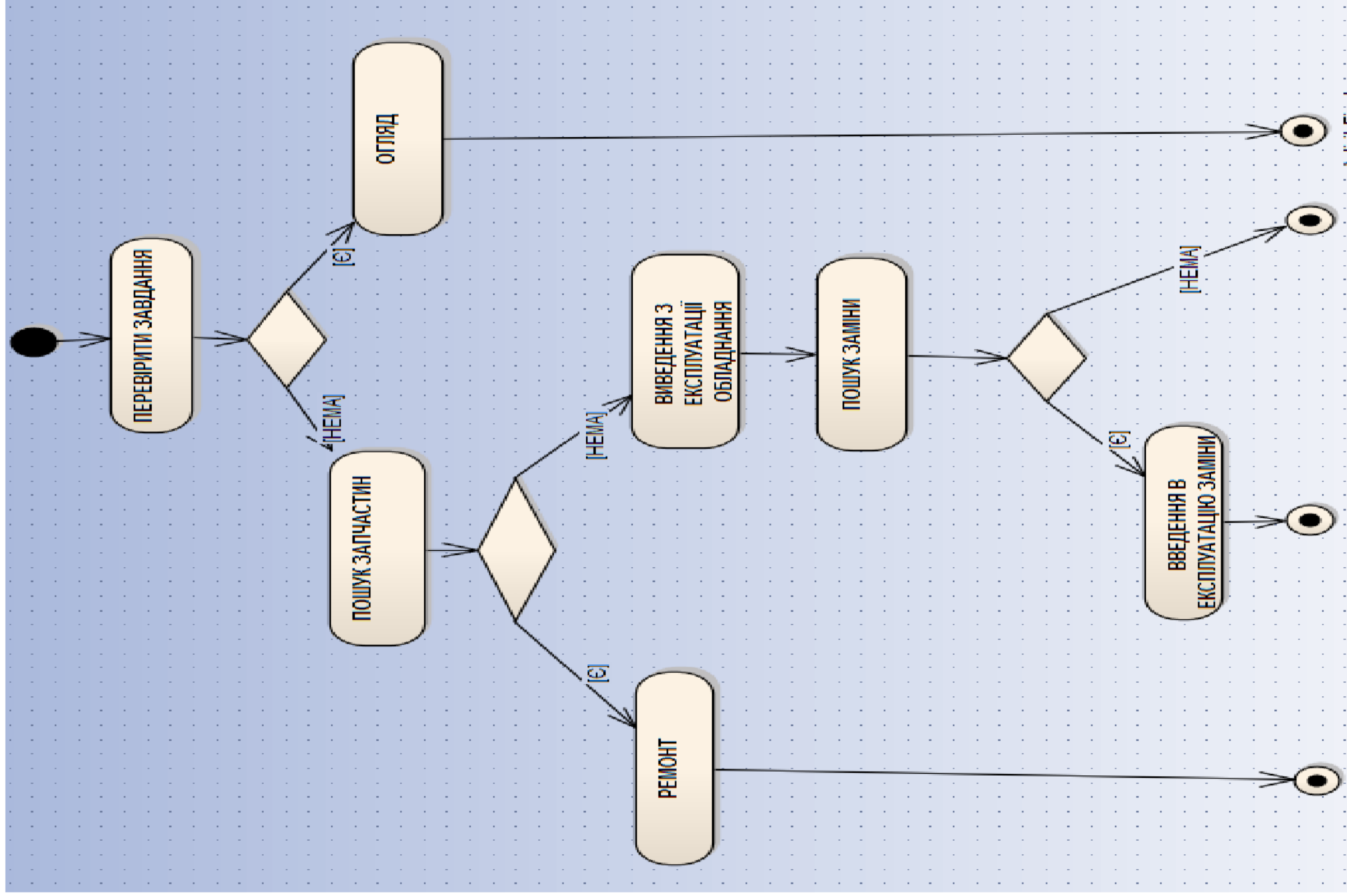
КІПЬКІСТЬ:

Buttons: ЗБЕРЕГТИ (Save), ВИХІД (Exit)

					ДП 6106.08.000 КЕ			
Зм.	Арк.	№ докум.	Підп.	Дата	креслення вигляду екранних форм	Лит.	Маса	Масштаб
Розроб.		Оробей Е.В.						
Перев.		Новікова П.А.						
Т. Кон.								
Н. Кон.		Телишева Т.О.			Автоматизована система управління технічним обслуговуванням виробничого обладнання	Аркуш		Аркушів
Затв.		Новікова П.А.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361		



					ДП 6106.06.000 ССП			
Зм.	Арк.	№ докум.	Підп.	Дата	схема структурна послідовності	Лист.	Маса	Масштаб
Розроб.		Оробей Е.В.						
Перев.		Новікова П.А.						
Т. Кон.						Аркуш	Аркушів	
Н. Кон.		Телишева Т.О.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361		
Затв.		Новікова П.А.						



ДП 6106.02.000 ССД

Зм.	Арк.	№ докум.	Підп.	Дата
Розроб.		Оробей Е.В.		
Перев.		Новикова П.А.		
Т. Кон.				
Н. Кон.		Телищева Т.О.		
Затв.		Новикова П.А.		

Лист.	Маса	Масштаб

Аркуш	Аркушів

схема структурна об'єктивності

Автоматизована система управління технічним обслуговуванням виробничого обладнання

КПІ ім. Іоанна Сікорського
Каф. АСОІУ
Гр. ІС-361

