

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Інститут інформаційних технологій та комп'ютерної інженерії
(повне найменування інституту, назва факультету (відділення))

Кафедра програмного забезпечення
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до бакалаврської дипломної роботи

бакалавр

(освітньо-кваліфікаційний рівень)

на тему: «Розробка програмного забезпечення для бази даних підприємства
з виробництва приладів обліку енергоносіїв»

Виконав: студент IV курсу, групи 1ПІ-10б

напряму підготовки

6.050103 – Програмна інженерія

(шифр і назва напряму підготовки, спеціальності)

Стрельбіцький М.Ю.

(прізвище та ініціали)

Керівник: доц., к.т.н. Майданюк В.П.

(прізвище та ініціали)

Рецензент: проф., д.т.н. Роїк О. М.

(прізвище та ініціали)

Вінницький національний технічний університет
Інститут інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Освітньо-кваліфікаційний рівень – бакалавр
Напрямок підготовки 6.050103 «Програмна інженерія»

УЗГОДЖЕНО

Директор ТОВ НВП «ГРЕМПС»

_____ В.М. Білоус

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, проф., д.т.н.

_____ А.М. Петух
“___” _____ 2014 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Стрельбіцькому Максиму Юрійовичу

1. Тема роботи: «Розробка програмного забезпечення для бази даних підприємства з виробництва приладів обліку енергоносіїв», керівник роботи: Майданюк Володимир Павлович, к.т.н., доцент затверджені наказом вищого навчального закладу від “___” _____ 20__ року №___
2. Строк подання студентом роботи _____
3. Вихідні дані до роботи: операційна система - Windows XP, Windows 7
мова програмування - C++
система керування БД - MySQL
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ; аналіз завдання і вибір методів його вирішення; розробка моделі бази даних та структури програмного продукту; розробка і тестування програмного продукту; охорона праці; висновки; додатки
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): мета, об'єкт та предмет дослідження; аналіз реляційних СКБД та вибір методу вирішення завдання; розробка ER-моделі бази даних; розробка структури програми; вибір мови програмування для реалізації програмного продукту; розробка програмних модулів: модуль авторизації; модуль баз відправлень, надходжень та повірок; інші модулі; результати тестування; охорона праці; результати обробки та висновки

6. Консультанти розділів проекту (роботи)

Роз-діл	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Майданюк В.П., доцент кафедри ПЗ		
4	Заюков І.В., доцент кафедри БЖД		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз розвитку баз даних	04.02.14	Вик.
2	Розробка моделі бази даних	06.02.14	Вик.
3	Розробка структури та інтерфейсу програми	28.02.14	Вик.
4	Розробка програмних модулів	18.04.14	Вик.
5	Тестування роботи програмного продукту	12.05.14	Вик.
6	Розробка керівництва користувача	15.05.14	Вик.
7	Розробка питань охорони праці	30.05.14	Вик.
8	Оформлення матеріалів до захисту БДР	09.06.14	Вик.

Студент

_____ (підпис)

Стрельбіцький М.Ю.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Майданюк В.П.

_____ (прізвище та ініціали)

АНОТАЦІЯ

У бакалаврській дипломній роботі розроблено програмний продукт для керування базою даних підприємства із виробництва та технічного обслуговування приладів обліку енергоносіїв. Програма дозволяє маніпулювати даними бази (додавати, редагувати та видаляти записи), друкувати звіти та технологічні карти пристроїв.

В ході виконання роботи проведено аналіз методів розв'язання задачі розробки, аналіз предметної галузі, розроблено структуру та модулі програмного продукту, а також виконано тестування роботи.

Також у бакалаврській дипломній роботі розглянуто питання охорони праці та забезпечення безпеки життєдіяльності людини.

АННОТАЦИЯ

В бакалаврской дипломной работе разработан программный продукт для управления базой данных предприятия по производству и техническому обслуживанию устройств учета энергоносителей. Программа дает возможность манипулировать данными базы (добавлять, редактировать и удалять записи), печатать отчеты и технологические карты устройств.

В ходе выполнения работы проведено анализ методов решения задачи разработки, анализ предметной области, разработана структура и модули программного продукта, а также произведено тестирование работы.

Также в бакалаврской дипломной работе рассмотрен вопрос охраны труда и обеспечения безопасности жизнедеятельности человека.

ANNOTATION

This paper addresses the software for managing the database of a company manufacturing and maintaining energy sources counters. The program allows manipulating database data (add, edit or remove records), printing reports and device process maps.

In the course of the paper was performed analysis of developing solving methods and specialization, was developed structure and program modules and was performed testing.

Issues of labor protection and human life protection also reviewed in the bachelor degree work.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ.....	8
1.1 Аналіз сучасного розвитку баз даних	8
1.2 Порівняльний аналіз із аналогічними програмними засобами	11
1.3 Аналіз та вибір методу розв'язання завдання	12
1.4 Постановка задачі розробки.....	14
1.5 Висновки	15
2 РОЗРОБКА МОДЕЛІ БАЗИ ДАНИХ ТА СТРУКТУРИ ПРОГРАМНОГО ПРОДУКТУ	16
2.1 Аналіз предметної галузі	16
2.2 Розробка ER-моделі бази даних.....	17
2.3 Розробка структури програмного продукту	20
2.4 Розробка інтерфейсу користувача	22
2.5 Висновки	25
3 РОЗРОБКА І ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	26
3.1 Аналіз та обґрунтування вибору мови програмування.....	26
3.2 Розробка програмних модулів	28
3.2.1 Модуль авторизації	29
3.2.2 Модуль "База відправлень"	30
3.2.3 Модуль "База надходжень"	31
3.2.4 Модуль "База перевірок"	32
3.2.5 Інші модулі	33
3.3 Тестування програмного продукту	34

	5
3.4 Керівництво користувача	40
3.5 Висновки	43
4 ОХОРОНА ПРАЦІ	44
4.1 Характеристика професії та приміщення	44
4.2 Аналіз стану охорони праці в приміщенні	47
4.3 Розробка заходів щодо покращення умов праці на робочому місці.....	53
4.4 Висновки	54
ВИСНОВКИ.....	56
ПЕРЕЛІК ПОСИЛАНЬ	57
Додаток А. Акт впровадження.....	59
Додаток Б. Лістинг вихідного коду програми.....	61
Додаток В. Ілюстративний матеріал до захисту бакалаврської роботи	98

ВСТУП

У бакалаврській дипломній роботі розглядається питання розробки програмного забезпечення для керування базою даних підприємства з виробництва та технічного обслуговування енергоносіїв. Завдання розробки баз даних на підприємствах різного роду є важливим етапом у впровадженні інформаційних технологій у процеси виробництва та обслуговування. Базы даних стають дедалі більш важливим та невід'ємним аспектом, особливо коли мова йде, наприклад, про облік виготовленої продукції за серійними номерами або про записи, пов'язані із ремонтом та технічним обслуговуванням. Централізованість та доступність даних значно полегшують процеси виробництва та допомагають компактно зберігати дані у цілісному вигляді.

Метою дослідження є автоматизація процесів обробки інформаційних ресурсів шляхом розробки та впровадження програми керування створеною базою даних.

Об'єктом дослідження є інформаційні технології та процеси обробки інформаційних ресурсів.

Предметом дослідження є моделі, структури та методи розробки автоматизованих систем обробки інформаційних ресурсів та створення баз даних.

Задачею розробки є створення структури бази даних, її початкове наповнення та власне розробка програмного забезпечення для керування нею.

У першому розділі аналізується поставлене завдання та пропонуються альтернативні методи його вирішення. Виконується аналіз сучасного розвитку у сфері баз даних та порівняльний аналіз аналогічних програмних продуктів. Аналізуються існуючі підходи до розв'язання поставленої задачі та обґрунтовується вибір одного із методів. Виконується постановка задачі розробки.

У другому розділі розробляється модель бази даних та структура програмного продукту. Зокрема, аналізується предметна галузь бази даних, а на її основі будується ER-модель майбутньої бази. Розглядається концепція структури програмного продукту та інтерфейсу користувача.

Третій розділ розглядає процес розробки і тестування програмного продукту. Виконується аналіз популярних мов програмування, обґрунтовується вибір однієї із них, а також обирається середовище розробки. Виконується безпосередня розробка продукту та його налагодження. Розробляється керівництво користувача та виконується тестування функціоналу.

В четвертому розділі розглядаються питання, пов'язані з охороною праці на підприємстві. Характеризується професія та приміщення у відділі програмного забезпечення. Аналізується стан охорони праці: стан мікроклімату, хімічного стану повітря, ефективність природного та штучного освітлення. Розробляються заходи щодо покращення умов праці на робочому місці.

Розробка є актуальною, оскільки підприємство давно потребує оновлення власного програмного забезпечення, і одним із важливим аспектів його роботи є ведення власної бази даних. Вона має також практичну цінність, оскільки розроблена програма впроваджується на цільовому підприємстві.

У ході виконання бакалаврської дипломної роботи використано 25 літературних джерел.

1 АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз сучасного розвитку баз даних

Сучасні темпи зростання обсягів інформації та комп'ютеризації в цілому дають поштовх для розвитку одного із фундаментальних понять в сфері комп'ютерних наук – баз даних.

Базу даних можна визначити як впорядкований набір взаємопов'язаних даних, що використовуються спільно та призначені для задоволення інформаційних потреб користувачів. Вона є зручним представленням великих обсягів даних, якими може зручно оперувати користувач і програміст [1].

Розвиток баз даних в сучасному їх розумінні розпочався у 1960-х рр. На той час інформація зберігалась в файлах, де кожен із них містив певні дані, а для охоплення усієї предметної області було необхідно декілька таких файлів. Така організація була досить незручною та неуніфікованою, а розробка програм викликала у програмістів великі труднощі.

В 1970-х рр. з'явилися перші теоретичні засади для створення реляційної моделі даних. Її перевагою стало мінімальне дублювання даних та виключення помилок, властивих для інших моделей. У цій моделі небажані таблиці можуть бути нормалізовані та розбиті на декілька взаємопов'язаних таблиць. Наприкінці 1970-х з'явилась перша реляційна система керування базами даних [2].

З середини 1980-х рр. почалось масове об'єднання комп'ютерів у мережі, що призвело до створення клієнт-серверної моделі, а також моделі із сумісним використанням файлів, яка дозволяла використовувати дорогі на той час принтери та дискові накопичувачі великої ємкості. Ці зміни торкнулись і баз даних; вони стимулювали розвиток цієї галузі у напрямку сумісного використання цих баз даних, а також створенню багатокористувацьких додатків для роботи із ними.

Продовжують активно розвиватись web-додатки баз даних, а також бази даних із використанням Internet-технологій. Розвитку також набувають об'єктно-

орієнтовані та розподілені бази даних. Це пов'язано із глобалізацією розвитку мережі та географічним розширенням меж використання самих баз даних.

Розвиток баз даних дав поштовх до створення систем керування базами даних – спеціальних програм, що забезпечують користувачам можливість створення, збереження, оновлення, пошук інформації та контролю доступу в базах даних. Основними тенденціями, які дали привід для проведення різних масштабних досліджень в області баз даних стали:

1. Експонентний ріст даних. Обсяг даних, у тому числі синтетичних, що генеруються автоматизованими системами, значно зріс. Збільшилося і число прикладних областей, в яких вимагається обробка великих обсягів даних. До таких областей тепер відносяться не тільки традиційні корпоративні програми, пошук у веб, але також і наукові дослідження, обробка природних мов, аналіз соціальних мереж тощо.
2. Значне ускладнення структур використовуваних даних. Прості види даних у вигляді чисел і символьних рядків стали доповнятися численною мультимедійною інформацією, просторовими, процедурними даними та великою кількістю інших складних форматів.
3. Широке поширення дешевих високопродуктивних апаратних засобів. Щорічно ми спостерігаємо зростання обчислювальних можливостей мікропроцесорів, збільшення ємності і зниження вартості доступних і зручних в експлуатації пристроїв дискової і оперативної пам'яті.
4. Активний розвиток засобів комунікації та «всесвітньої павутини» World Wide Web. WWW стає єдиним інформаційним середовищем, що пронизує весь світ і об'єднує величезне число користувачів та електронних пристроїв.
5. Поява нових важливих областей застосування СКБД. У першу чергу, це пов'язано з інтелектуальним аналізом даних, сховищами даних, а останнім часом – з паралельними обчисленнями і хмарними технологіями [3].

На сучасному етапі розвитку систем керування базами даних можна виділити такі особливості:

- підтримка повної реляційної моделі, а саме:
 - структурної цілісності* – допустимими є тільки дані, представлені у вигляді відносин реляційної моделі;
 - мовної цілісності*, тобто мов маніпулювання даними високого рівня (в основному SQL) ;
 - цілісності посилань*, контролю за дотриманням цілісності протягом всього часу функціонування системи і гарантій неможливості з боку СКБД порушити ці обмеження.
- більшість сучасних СКБД розраховані на кросплатформену архітектуру, тобто вони можуть працювати на комп'ютерах з різною архітектурою і під різними операційними системами, при цьому для користувачів доступ до даних, керованим СКБД на різних платформах, практично однаковий;
- необхідність підтримки багатокористувацької роботи з базою даних і можливість децентралізованого зберігання даних сприяють розвитку засобів адміністрування БД з реалізацією загальної концепції засобів захисту даних;
- потреба в нових реалізаціях викликала створення серйозних теоретичних праць з оптимізації реалізацій розподілених багатокористувацьких БД і роботі з розподіленими транзакціями і запитами з впровадженням отриманих результатів у вигляді комерційних СКБД;
- розробка ряду стандартів в рамках мов опису і маніпулювання даними починаючи з SQL89, SQL92, SQL99 і технологій по обміну даними між різними СУБД , до яких можна віднести і протокол ODBC (Open DataBase Connectivity), запропонований фірмою Microsoft.
- початок робіт, пов'язаних з концепцією об'єктно-орієнтованих БД – ООБД.

Прикладами сучасних реляційних СКБД можна вважати MS Access і всі сучасні сервери баз даних Oracle, MS SQL Server, MySQL, SQLBase та багато інших, яких зараз налічується кілька десятків [4].

1.2 Порівняльний аналіз із аналогічними програмними засобами

Сучасні бази даних використовуються у багатьох сферах, а функціонал програмного забезпечення для роботи із ними варіюється залежно від потреб користувачів. Оскільки авторська програма розроблялась виключно для використання в межах певного підприємства, її аналоги знайти важко або практично неможливо. Тим не менш, розглянемо деякі програмні продукти для керування базами даних у своїх сферах, функціонал яких буде подібним до авторської розробки.

Прикладами таких розробок є велика кількість українських та іноземних розробок. Зокрема, можна виділити розробки «Касир TechnoPOS» та «Система RecOrder» виробництва ТОВ «ТЕХНО+» [5], систему автоматизації роботи метеорологічної служби від компанії Альтсофт [6], базу даних фірми АСТОР із технічного обслуговування електричних мереж [7], тощо.

Кожна із перерахованих програм унікальна та має власну чітко визначену предметну галузь, в якій вона застосовується. Проте, більшість програмних продуктів, пов'язаних із базами даних, надають типовий функціонал, а саме:

- додавання нових записів;
- редагування існуючих записів;
- видалення некоректних записів;
- представлення одного набору даних у різних поданнях (формах, таблицях, тощо);
- швидкий пошук та фільтрація даних;
- виведення звітів на екран та друк.

Типовим вимогам відповідатиме і програмний продукт, розроблюваний у бакалаврській дипломній роботі. Крім переліченого функціоналу аналогів, він передбачає також базові можливості налаштування інтерфейсу та роботу в декількох режимах.

1.3 Аналіз та вибір методу розв'язання завдання

Для розв'язання задачі розробки програми керування базою даних необхідно визначитись із типом СКБД, яка буде використовуватись у програмі. Для розробки бази даних буде застосовано реляційну модель. Дамо визначення СКБД та порівняємо деякі популярні їх види.

СКБД – це складна програмна система для накопичення та маніпулювання даними, що представляють інтерес для користувача. Кожній прикладній програмі СКБД надає інтерфейс для роботи з базою даних та має засоби безпосереднього доступу до неї. Таким чином, СКБД відіграє центральну роль в функціонуванні баз даних.

До основних можливостей сучасних СКБД можна віднести такі:

- контроль за надлишковістю даних;
- несуперечливість даних;
- підтримка цілісності бази даних;
- незалежність прикладних програм від даних;
- спільне використання даних;
- підвищений рівень безпеки, тощо.

Ключовими вимогами до сучасних СКБД є їх швидкодія, захищеність, розмежування рівнів доступу до даних, наявність механізмів резервного копіювання, тощо [3]. На сьогодні розроблено дуже велику кількість різноманітних СКБД. Для реалізації програмного продукту доцільно скористатись реляційною СКБД, і для порівняння розглянемо деякі популярні із них: SQLite, MySQL, PostgreSQL.

SQLite легко вбудовується у програмні додатки. Система базується на файлах, тому надає досить широкий набір для роботи із нею, порівняно із мережевими СКБД. При роботі із цією СКБД відбувається пряме звернення до файлів, у яких зберігаються дані, замість портів та сокетів у мережеских СКБД. Саме завдяки цьому SQLite дуже швидка та потужна [8].

Переваги SQLite:

- файлова структура. Вся база даних зберігається в одному файлі, тому її легко переносити на різні комп'ютери;
- підтримка стандартів SQL;
- легкість у розробці та тестуванні.

Недоліки SQLite:

- відсутність системи користувачів;
- відсутність можливості збільшення продуктивності.

MySQL – найрозповсюдженіша повноцінна серверна СКБД. MySQL дуже функціональна, вільно розповсюджувана СКБД, яка успішно працює як із окремими додатками, так і з сайтами та web-додатками. Вона легка у вивченні, оскільки у неї зручна документація та їй присвячена велика кількість джерел, які стосуються СКБД та баз даних в цілому. Завдяки популярності MySQL існує велика кількість різноманітних плагінів та розширень, які полегшують роботу із системою.

Незважаючи на те, що у ній не реалізовано весь функціонал SQL, MySQL пропонує доволі велику кількість інструментів для розробки додатків. Оскільки це серверна СКБД, то, на відміну від SQLite, доступ до даних відбувається через служби MySQL [8].

Переваги MySQL:

- простота в роботі;
- багатий функціонал;
- масштабованість;
- висока продуктивність.

Недоліки MySQL:

- обмеження функціональності, необхідні для окремих задач;
- проблеми з надійністю;
- порівняно невисокі темпи розробки додатків.

PostgreSQL є найбільш професійною із розглянутих СКБД. Вона вільно розповсюджується та максимально відповідає стандартам SQL. Від інших СКБД вона відрізняється підтримкою об'єктно-орієнтованого та реляційного підходів до баз

даних. Завдяки потужним технологіям PostgreSQL дуже продуктивна. Її можна доповнювати власними процедурами, щоб полегшити використання повторюваних операцій.

Хоча PostgreSQL і не настільки популярна, як MySQL, але існує доволі велика кількість додатків, які полегшують роботу із PostgreSQL, зважаючи на потужність її функціоналу [8].

Переваги PostgreSQL:

- відповідність стандартам та відкритий вихідний код;
- велика кількість розробників;
- можливість розширення функціоналу;
- об'єктна орієнтованість.

Недоліки PostgreSQL:

- недостатня швидкодія;
- низька популярність;
- складність у вивченні.

Зважаючи на перелічені особливості, для реалізації в авторській програмі буде використано СКБД MySQL, яка відповідає вимогам до програми та в найбільш повній мірі дозволить реалізувати необхідний функціонал.

1.4 Постановка задачі розробки

Оскільки завданням бакалаврської дипломної роботи є розробка програми керування базою даних підприємства, можна виділити такі задачі для виконання:

- розглянути можливість застосування бази даних у предметній галузі;
- розробити інфологічну модель бази даних;
- розробити структуру програми;
- розробити інтерфейс програми;
- порівняти можливості мов програмування для реалізації програми та обрати одну із них для використання;
- розробити програму;

- розробити керівництво користувача;
- провести тестування та виправлення помилок;
- розглянути питання охорони праці.

1.5 Висновки

У першому розділі розглянуто питання розвитку баз даних, виконано короткий огляд аналогів, обрано метод вирішення поставленого завдання та виконано постановку задачі.

Бази даних – складні та масивні структури, які використовуються для зберігання великих обсягів інформації. У зв'язку із розвитком інформаційних технологій та експонентним зростанням обсягів інформації, бази даних набули широкого вжитку та розвитку, який продовжується і досі. Одним із результатів цього стало створення систем керування базами даних (СКБД), які використовуються у програмному забезпеченні, пов'язаному із базами даних. Зокрема, одна із СКБД, MySQL, буде використана при розробці програмного продукту, який розглядається у бакалаврській дипломній роботі.

2 РОЗРОБКА МОДЕЛІ БАЗИ ДАНИХ ТА СТРУКТУРИ ПРОГРАМНОГО ПРОДУКТУ

2.1 Аналіз предметної галузі

Програма, яка розробляється у бакалаврській дипломній роботі, оперуватиме базою даних із інформацією щодо пристроїв, які реалізовує підприємство, а також тих, які проходять на ньому обслуговування.

Підприємство реалізує велику кількість найменувань приладів, зокрема комплекти приладів для комерційного і технологічного обліку спожитого природного газу, теплової енергії, повітря й води із застосуванням мікропроцесорних обчислювачів та коректорів об'єму газу "Універсал", які на сучасному рівні вирішують завдання обліку на промислових підприємствах, забезпечуючи високу точність, раціональність і економічність режиму споживання енергоносіїв [9].

Пристрої типу "Універсал" зберігатимуться у список реалізованих пристроїв. Вони мають свій набір технологічних параметрів (серійний номер, тип, рік випуску, версія ПЗ, тип плати, частота роботи, тип пам'яті та процесора, тощо), інформацію про отримувача пристрою та технічні примітки.

У переліку надходжень зберігаються усі пристрої, які надходять до фірми на обслуговування: обчислювачі та коректори, бар'єри іскрозахисту, датчики температури, тощо. Про них зберігатиметься основна технологічна інформація, а також інформація про причини надходження (ремонт, повірка, модернізація, повернення, тощо), дані про ремонтні роботи (перелік кодів та вартості послуг), бухгалтерські дані (реквізити відправника та отримувача, номери рахунків, статус оплати) та примітки.

Згідно із вимогами користувача, програма надаватиме широкі можливості щодо оперування даними (додавання, видалення, редагування, клонування), створення звітів (як загальних, так і спеціалізованих) та фільтрації даних за певними параметрами.

2.2 Розробка ER-моделі бази даних

Для представлення структури бази даних необхідно розробити її інформаційну структуру (або концептуальну схему).

В звичайних випадках для побудови концептуальної схеми використовуються традиційні методи агрегації та узагальнення. При агрегації декілька інформаційних об'єктів (елементів даних) об'єднується в один у відповідності з семантичними зв'язками між об'єктами. При побудові інфологічних моделей використовують мову ER-діаграм ("сутність-зв'язок"). В них сутності зображуються позначеними прямокутниками, асоціації - позначеними ромбами або шестикутниками, атрибути - позначеними прямокутниками, а зв'язки між ними - ненаправленими ребрами, над якими може проставлятися ступінь і необхідне пояснення.

Головними поняттями ER-моделі є сутність, зв'язок і атрибут. У діаграмах ER-моделі сутність подається у вигляді прямокутника, що містить ім'я сутності. При цьому ім'я сутності - це ім'я типу, а не деякого конкретного екземпляра цього типу. Кожний екземпляр сутності повинен відрізнятися від будь-якого іншого екземпляра тієї ж сутності (ця вимога до повної міри аналогічна вимозі відсутності кортежів-дублікатів у реляційних таблицях) [10].

У розроблюваній базі даних можна виділити такий набір сутностей: запис, місто, причина надходження, прейскурант, коди робіт, користувач.

Розглянемо, яким чином взаємопов'язані між собою ці сутності.

Зв'язок Місто – Запис. Кожному запису відповідає певне місто: пристрій надсилається або надходить за однією адресою. Тип зв'язку у цьому випадку – 1:N, клас належності сутності Місто необов'язковий (рис. 2.1).

Зв'язок Запис – Причина надходження. Записам відповідає причина надходження у випадку їх обслуговування. Тип зв'язку – N:1, клас належності обох сутностей необов'язковий (рис. 2.2).

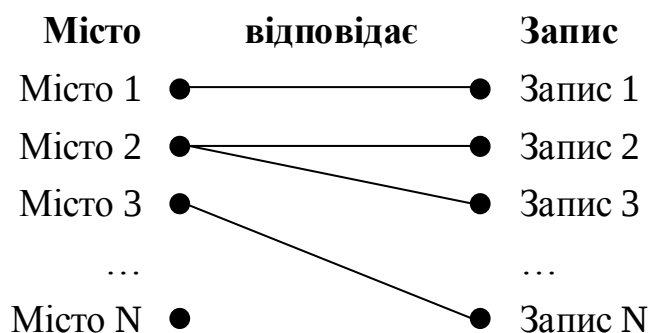


Рисунок 2.1 – Діаграма екземплярів сутей Місто і Запис

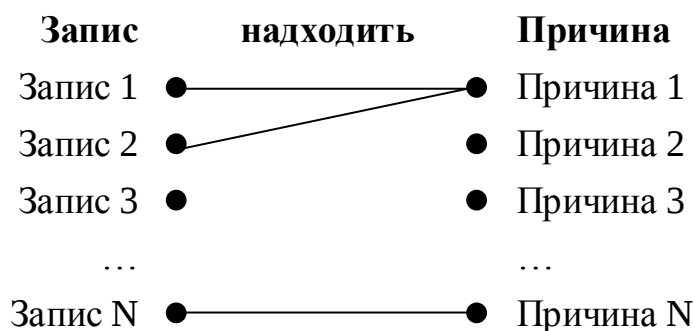


Рисунок 2.2 – Діаграма екземплярів сутей Запис і Причина надходження

Зв'язок Запис – Коді робіт. У випадку ремонтних робіт кожному запису відповідає один або декілька кодів робіт. Тип зв'язку – 1:N, клас належності сутності *Запис* необов'язковий (рис. 2.3).

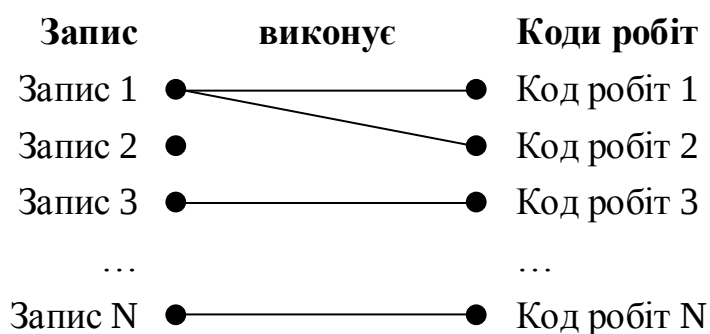


Рисунок 2.3 – Діаграма екземплярів сутей Запис і Коді робіт

Зв'язок Запис - Користувач. У процесі редагування та/або додавання записів додається інформація користувачів, які зробили ту чи іншу дію. Тип зв'язку – M:N, клас належності сутності Користувач необов'язковий (рис. 2.4).

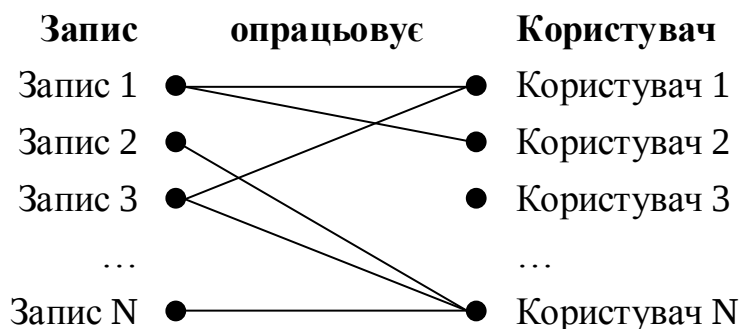


Рисунок 2.4 – Діаграма екземплярів сутей Запис і Користувач

Зв'язок Прейскурант – Коди робіт. Щоб зберігати для одного запису набір з декількох кодів та відповідних їм цін, таблиця прейскурантів робіт та набір їх наборів для кожного із записів пов'язані. Тип зв'язку – 1:N. Клас належності сутності *Прейскурант* необов'язковий (рис. 2.5).

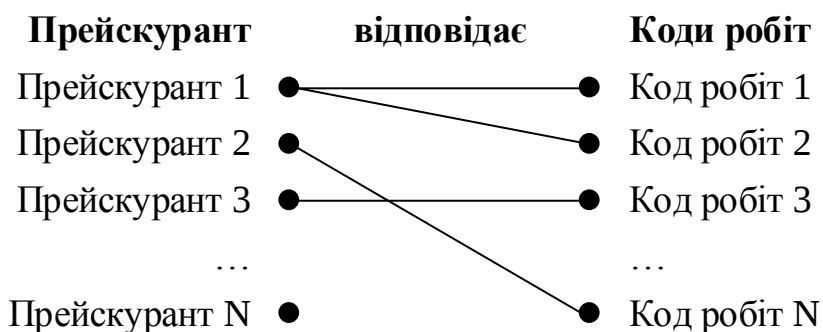


Рисунок 2.5 – Діаграма екземплярів сутей Прейскурант і Коды робіт

Повний перелік зв'язків наведено в табл. 2.1.

Таблиця 2.1 – Характеристики зв'язків сутностей

Назва сутності 1	Назва сутності 2	Тип зв'язку	Назва зв'язку	Клас належності
Місто	Запис	1:N	відповідає	Необ.; Обов.
Запис	Причина	N:1	надходить	Необ.; Необ.
Запис	Коди робіт	1:N	виконує	Необ.; Обов.
Запис	Користувач	M:N	опрацьовує	Обов.; Необ.
Прейскурант	Коди робіт	1:N	відповідає	Необ.; Обов.

За допомогою отриманих даних будуємо результуючу ER-модель (рис. 2.6).

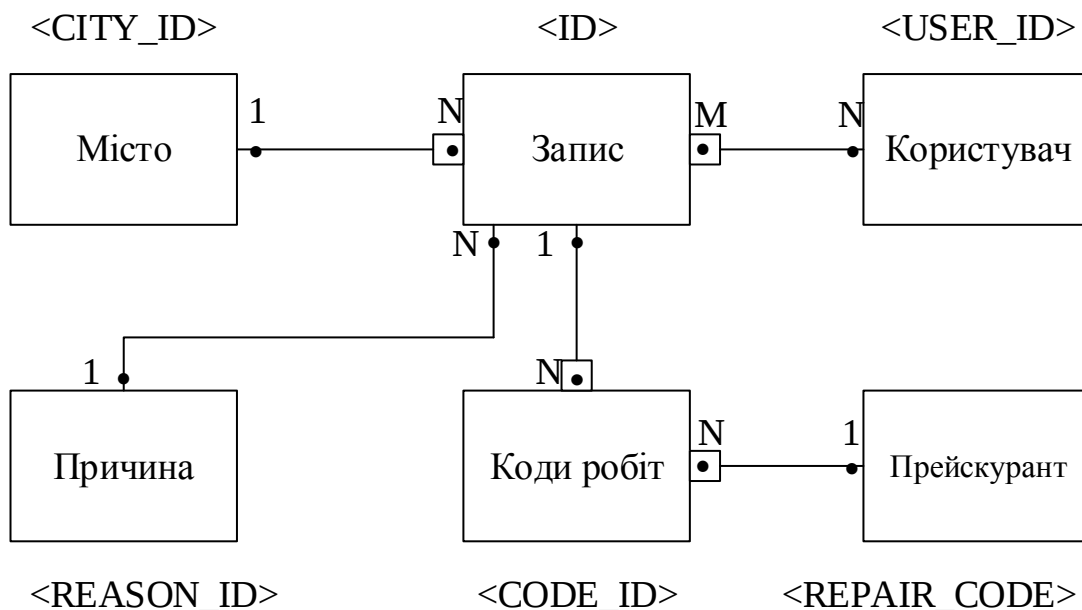


Рисунок 2.6 – ER-модель предметної області

2.3 Розробка структури програмного продукту

Для реалізації функціоналу програмного продукту його доцільно розділити на модулі, які відповідають за ті чи інші властивості та функції. Передбачається, зокрема, розробка таких модулів та форм.

Модуль авторизації передбачає введення від користувача введення його облікової інформації, якщо його обліковий запис уже створений, або ж створення нового облікового запису. В подальшому дані користувача запам'ятовуються у реєстрі, і подальшої потреби у введенні логіна та паролю немає. Додаткові форми: створення облікового запису користувача.

Модуль "База відправлень" зберігає інформацію про пристрої власного виробництва фірми. Дозволяє користувачеві додавати, редагувати та видаляти пристрої, створювати технічний коментар. Додаткові форми: додавання / клонування / редагування записів; історія ремонтних робіт; розширені технічні примітки.

Модуль "База надходжень" зберігає інформацію про пристрої, які обслуговуються фірмою. Дозволяє користувачам додавати нові пристрої, обирати із

уже існуючих, додавати інформацію про проведені ремонтні роботи, виписані рахунки, реквізити відправників/отримувачів, тощо. Є можливість перегляду історії повірок. Додаткові форми: додавання / клонування / редагування записів; інформація про ремонтні роботи; бухгалтерська інформація; інформація про отримання/відправку; історія повірок.

Модуль "База повірок" зберігає інформацію про прилади в процесі роботи, які потребують перевірки. Дані безпосередньо пов'язані із базою надходжень. Додаткові форми: загальна інформація про прилади; пошук за датою.

Модуль фільтрації даних дозволяє фільтрувати дані відповідно до введених користувачем критеріїв.

Модуль друку передбачає друк звітів по таблицях, а також друк бирок для пристроїв, які перебувають на обслуговуванні.

Модуль шрифтів дозволяє змінювати розмір шрифту у таблиці (звичайний/збільшений/великий).

Загальний взаємозв'язок між модулями проілюстровано на рис. 2.1.

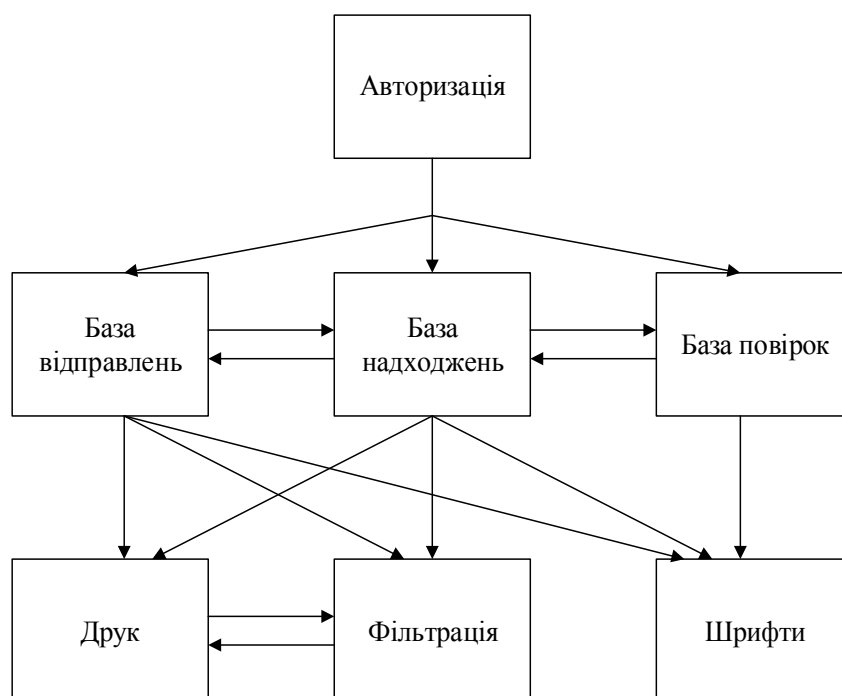


Рисунок 2.7 – Структура програми

2.4 Розробка інтерфейсу користувача

Інтерфейс – зовнішня оболонка додатку разом з програмами управління доступом та іншими прихованими від користувача механізмами управління, яка дає можливість працювати з документами, даними та іншою інформацією, що зберігається в комп'ютері або за його межами. Головна мета будь-якої програми – забезпечити максимальну зручність і ефективність роботи з інформацією: документами, базами даних, графікою та зображеннями. Добре розроблений інтерфейс гарантує зручність роботи з додатком [11].

Для реалізації у авторській програмі було обрано WIMP-інтерфейс, який полягає у діалозі користувача із системою за допомогою графічних елементів: меню, вікон, тощо.

Загалом інтерфейс програми складається із різних вікон, але всі дії розпочинаються у головному вікні. Воно складається із головного меню, панелі інструментів (кнопок), панелі фільтрації, власне таблиці, пов'язаної із базою даних, та рядку стану. Схематично його можна показати так (рис. 2.8).

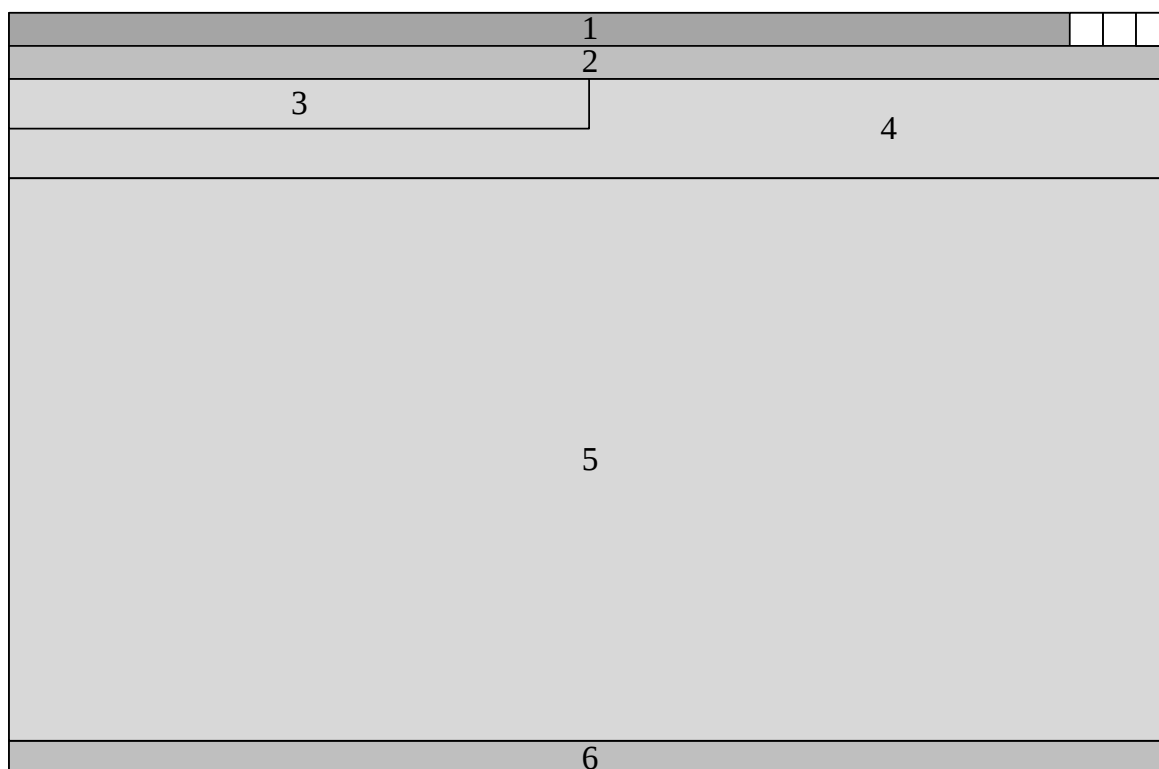


Рисунок 2.8 – Прототип інтерфейсу головного вікна програми

Умовні позначення:

- 1 – панель керування вікном;
- 2 – панель меню;
- 3 – панель інструментів;
- 4 – панель фільтрації даних;
- 5 – робоча область (таблиця даних);
- 6 – рядок стану.

Із головного меню можна обрати режим роботи (база відправлень, база надходжень, база перевірок), виконати налаштування щодо переліку відображуваних полів, а також скоригувати розмір шрифтів у таблиці.

При натисненні на кнопки панелі інструментів з'являються дочірні вікна або діалоги, залежно від функції, яка виконується. Так, при додаванні нового пристрою з'являється форма, зразок якої показано на рис. 2.9.

Рисунок 2.9 – Форма додавання пристрою

Оскільки програма передбачає можливість друку звітів та технологічних карт, для них передбачені окремі форми. Вони дозволяють зробити попередній перегляд перед друком (рис. 2.10, 2.11).

Друк звіту

80%

Close

Дата надх.	Дата відпр.	Клас	Тип	Номер	Рік вип.	Причина надх.	Область	Місто	Відправник	Версія / діап.	Частота / довжина
18.01.14		Універсал	02	100001	2014	Повірка	Вінницька	Вінниця	Петренко І.О.	12.01	
24.01.14	03.02.14	Універсал	02	100004	2014	Ремонт-повірка	Вінницька	Бар	Максимович О.В.	12.01	
24.01.14		Універсал	01	100003	2014	Повірка	Вінницька	Бар	Максимович О.В.	12.01	
25.01.14	23.03.14	БІ	02-01	11325	2014	Повернення	Черкаська	Черкаси	ТОВ Черкашаночка	6.01	
23.03.14	23.03.14	Aplisens	01	1234	2014	Гарантія	Кіровоградська	Кіровоград	Петренко І.О.	12.01	7,37280
23.03.14	23.03.14	Aplisens	01	10320	2014	Гарантія	Кіровоградська	Кіровоград	Петренко І.О.	12.01	7,37280
23.03.14	23.03.14	Універсал	01	1082	2014	Повірка	Хмельницька	Баранівка	Максимович О.В.	12.01	7,37280
23.03.14	23.03.14	Універсал	02	394572	2014	Заміна	Вінницька	Козятин	Максимович О.В.		

Page 1 of 1

Рисунок 2.10 – Друк звіту

Друк технологічних карт

100%

Close

Пишіть розбірливо - поважайте працю інших!

Дата відправки: "___" "___" 20__

Кому: Максимович О.В.
Область: Хмельницька
Місто: Баранівка

01 № 1082 рік вип. 2014 дата надх. 23.03.14

Приймає: Стрельбицький М.Ю. Дата введення: 5.02.14, 09:29

Причина надходження: Повірка

№	Операція	ОК	Виконав	Дата (д.м.м.)	Примітка
1.	Клавіатура, табло				
2.	Наладка бл. живл.				
3.	Наладка годинника				
4.	Оживлення, дата				
5.	Частота				7,37280
6.	Перевірка АБ12м				
7.	Калібрування				
8.	Точна дата і час				
9.	Датчас повної готовності до прогону: "___" "___" 20__				
10.	Перевірка базису				
11.	Інтерфейс RS485-I				
12.	Інтерфейс RS232				модем ☐
13.	Сигнал DTR				
14.	Акум. (не менше 3)				
15.	Опер. програмув.				
16.	Прогон 96 годин				

При-мтки

Модернізація / ремонт:

☒ Документи Версія: 12.01 Плата: Память:

☐ К-ть роз'ємів: ☐ інше:

☐ NXP ☐ mega128 ☐ mega163

Дата друку: 13.04.14, 20:21

Пишіть розбірливо - поважайте працю інших!

Дата відправки: "___" "___" 20__

Кому: Забужко О.І.
Область: Вінницька
Місто: Вінниця

02 № 12 рік вип. 2014 дата надх. 13.04.14

Приймає: Стрельбицький М.Ю. Дата введення: 13.04.14, 20:01

Причина надходження: Повірка

№	Операція	ОК	Виконав	Дата (д.м.м.)	Примітка
1.	Клавіатура, табло				
2.	Наладка бл. живл.				
3.	Наладка годинника				
4.	Оживлення, дата				
5.	Частота				7,37280
6.	Перевірка АБ12м				
7.	Калібрування				
8.	Точна дата і час				
9.	Датчас повної готовності до прогону: "___" "___" 20__				
10.	Перевірка базису				
11.	Інтерфейс RS485-I				
12.	Інтерфейс RS232				модем ☐
13.	Сигнал DTR				
14.	Акум. (не менше 3)				
15.	Опер. програмув.				
16.	Прогон 96 годин				

При-мтки

Модернізація / ремонт:

☒ Документи Версія: 12.03 Плата: NXP Память: В

☐ К-ть роз'ємів: 3 ☐ інше:

☐ NXP ☐ mega128 ☐ mega163

Дата друку: 13.04.14, 20:21

Page 1 of 1

Рисунок 2.11 – Друк технологічних карт

Решта форм подібні за своєю структурою, а поля варіюються залежно від їх змісту. У разі виникнення помилок або діалогу з користувачем, з'являються стандартні діалогові вікна.

2.5 Висновки

У другому розділі розглядається модель бази даних, якою керує програма, а також структура програми та інтерфейс користувача. Відповідно до предметної галузі, у базі даних створюється відповідний набір полів, який відповідає фізичним характеристикам приладів та додатковим даним.

Для деталізації розробки програмний продукт розділено на модулі, кожен із яких відповідає за певний функціонал. Розроблюваний інтерфейс відповідає за зручність та швидкість доступу до даних користувачем, а також за використання функціоналу програми у повній мірі.

3 РОЗРОБКА І ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

3.1 Аналіз та обґрунтування вибору мови програмування

Для реалізації програми доцільно використати об'єктно-орієнтовану мову програмування. До найпопулярніших із них належать C++, C# та Java.

C++ – високорівнева мова з підтримкою об'єктно-орієнтованої, узагальненої та процедурної парадигм програмування. Розроблена Б'ярном Страуструпом у 1979 році. Стандартна бібліотека C++ включає стандартну бібліотеку Cі з невеликими змінами, які роблять її відповіднішою для мови C++. Інша велика частина бібліотеки C++ заснована на Стандартній Бібліотеці Шаблонів (STL). Вона надає такі важливі інструменти, як контейнери (наприклад, вектори і списки) та ітератори (узагальнені вказівники), що надають доступ до цих контейнерів як до масивів. Крім того, STL дозволяє схожим чином працювати і з іншими типами контейнерів, наприклад, асоціативними списками, стеками, чергами. Використовуючи шаблони, можна писати узагальнені алгоритми, здатні працювати з будь-якими контейнерами або послідовностями, доступ до членів яких забезпечують ітератори [12].

C# – об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft. Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Перейнявши багато що від своїх попередників, C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем: так, C# не підтримує множинне спадкування класів (на відміну від C++) або виведення типів (на відміну Haskell) [13].

Java – об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році як основний компонент платформи Java. Синтаксис мови багато в чому походить від C та C++. У офіційній реалізації, Java програми компілюються у байткод, який при виконанні інтерпретується віртуальною машиною

для конкретної платформи. Sun Microsystems надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License. Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім, Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування [14].

Оцінюючи вищеописані особливості, виділимо деякі переваги використання C++ при розробці програми:

- швидкість роботи програм на C++, порівняно із C# та Java, найвища;
- наявність стандартної бібліотеки STL дозволяє оперувати складними структурами даних;
- наявність зручних інструментів для роботи із реєстром та файловою системою;
- реалізація СКБД MySQL виконана мовами C та C++;
- наявність готових інтерфейсів для швидкої та зручної роботи із базою даних.

Порівняно із C++, мови Java та C# мають і певні недоліки:

- у Java ціною кросплатформеності є вимога наявності на комп'ютері віртуальної Java-машини, що приводить до уповільнення обчислень і практичної неможливості використання нових можливостей апаратної архітектури;
- збірка сміття призводить до втрати ефективності;
- використання вказівників у багатьох випадках є потужним та необхідним засобом, відсутнім у Java та C#.

Загальні особливості мов порівнюються у табл. 3.1 [15].

Таблиця 3.1 – Порівняння мов програмування C++, C# та Java

	C++	C#	Java
Рік створення	1983	2000	1995
Сфера застосування	прикладне та системне ПЗ	бізнес-додатки, клієнтські та серверні застосунки, веб	прикладне ПЗ, бізнес-додатки, клієнтські та серверні застосунки, веб
Стандарти	1998, ISO/IEC 1998, ISO/IEC 2003, ISO/IEC 2011	2000, ECMA, ISO	Java Language Specification (де-факто)
Парадигми програмування:			
- імперативна	+	+	+
- об'єктно-орієнтована	+	+	+
- функціональна	+	+	-
- процедурна	+	-	-
- узагальнена	+	+	+
- рефлексивна	-	+	+
- подійно-орієнтована	-	+	-
Захищеність типів	-	-	+
Типізація	явна	неявна / явна	явна
Сумісність типів	номінальна	номінальна	номінальна
Перевірка типів	статична	статична	статична
Захищеність вводу/виводу	-	+	+

Перевагою мови C++ є її швидкодія; вихідний код СКБД MySQL написаний саме мовою C++. Крім того, для мови C++ існують готові засоби для того, щоб розроблюваний продукт міг напряду звертатись до MySQL-сервера. Зокрема, для цього використовується компонент FireDAC у середовищі Embarcadero RAD Studio, яке дозволяє створювати програми мовою C++. Враховуючи перераховані вище переваги, для розробки програми буде використано мову програмування C++.

3.2 Розробка програмних модулів

Як було наведено в розділі 2, програма, розроблювана у бакалаврській дипломній роботі, має модульну структуру. При розробці використовується мова програмування C++ та середовище програмування Embarcadero C++ Builder. Для

підключення до бази даних використовується компонент FireDAC, таблиця даних головного вікна використовує можливості компонента TMS AdvStringGrid, для друку звітів обрано компонент FastReport, а компонент AlphaControls відповідає за кнопки графічного інтерфейсу.

Опишемо детально розробку та принцип роботи кожного із модулів.

3.2.1 Модуль авторизації

Модуль авторизації використовується при першому запуску програми. Він представляє собою діалог з користувачем, який повинен надати свої облікові дані для початку роботи з програмою (рис. 3.1).

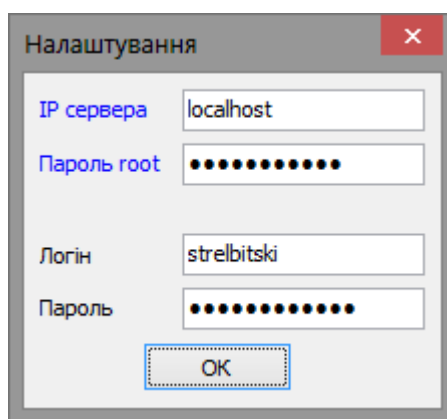


Рисунок 3.1 – Вікно модуля авторизації

При першому запуску за допомогою запитів на сервері та у базі даних створюється користувач із необхідними правами доступу:

```
GRANT ALL PRIVILEGES ON grempisdb.* TO user@localhost
IDENTIFIED BY password WITH GRANT OPTION;
```

```
GRANT ALL PRIVILEGES ON grempisdb.* TO user@% IDENTIFIED BY
password WITH GRANT OPTION;
```

```
INSERT INTO users VALUE (NULL, "user", "name").
```

Інформація про користувача записується в реєстр, і при повторному запуску програми вона зчитує його дані з гілки HKEY_CURRENT_USER\Software\GREMPIS\

Database. Після успішного зчитування інформації про з'єднання за допомогою компонента відбувається завантаження головного вікна програми. Відбувається спроба читання налаштувань інтерфейсу з файлу *interface.ini*, а якщо він ще не створений, встановлюються стандартні значення. Перевіряється набір полів, позиція вікна, розмір шрифтів, обраний режим. Крім цього, створюються та ініціалізуються постійні списки: список міст та областей України та список клієнтів, зареєстрованих у базі даних.

3.2.2 Модуль "База відправлень"

Модуль "База відправлень" використовується для обліку пристроїв власного виробництва, які реалізуються споживачам. Головне вікно у цьому режимі показано на рис. 3.2. Модуль надає такий функціонал:

- додавання нових записів;
- створення нових записів на основі існуючих, або клонування параметрів;
- редагування записів;
- видалення записів;
- перегляд історії ремонтних робіт та встановлення технічних коментарів;
- створення міток.

Номер	Тип	Версія	Плата	Проц.	Пам.	X	Частота	Рік вип.	Дата відправки	Область	Місто	Перший встан.	Поточний встан.	Дата редагування	Редактор
8	02							2014	08.03.14	Вінницька	Козятин	Максимович О.В.	Максимович О.В.	08.03.14 15:01	Стрельбицький М.Ю.
9	02							2014	08.03.14	Вінницька	Козятин	Максимович О.В.	Максимович О.В.	08.03.14 15:01	Стрельбицький М.Ю.
10	02							2014	08.03.14	Вінницька	Козятин	Максимович О.В.	Максимович О.В.	08.03.14 15:01	Стрельбицький М.Ю.
11	02							2014	08.03.14	Вінницька	Козятин	Максимович О.В.	Максимович О.В.	08.03.14 15:01	Стрельбицький М.Ю.
12	02							2014	08.03.14	Вінницька	Козятин	Максимович О.В.	Максимович О.В.	08.03.14 15:01	Стрельбицький М.Ю.
13	02							2014	08.03.14	Вінницька	Козятин	Максимович О.В.	Максимович О.В.	08.03.14 15:01	Стрельбицький М.Ю.
14	02							2014	08.03.14	Вінницька	Козятин	Максимович О.В.	Максимович О.В.	08.03.14 15:01	Стрельбицький М.Ю.
15	02							2014	08.03.14	Вінницька	Козятин	Максимович О.В.	Максимович О.В.	08.03.14 15:01	Стрельбицький М.Ю.
16	02							2014	08.03.14	Вінницька	Козятин	Максимович О.В.	Максимович О.В.	08.03.14 15:01	Стрельбицький М.Ю.
17	02							2014	08.03.14	Вінницька	Козятин	Максимович О.В.	Максимович О.В.	08.03.14 15:01	Стрельбицький М.Ю.
18	02							2014	08.03.14	Вінницька	Козятин	Максимович О.В.	Максимович О.В.	08.03.14 15:01	Стрельбицький М.Ю.
19	02							2014	08.03.14	Вінницька	Козятин	Максимович О.В.	Максимович О.В.	08.03.14 15:02	Стрельбицький М.Ю.
20	02							2014	08.03.14	Вінницька	Козятин	Максимович О.В.	Максимович О.В.	08.03.14 15:02	Стрельбицький М.Ю.

Користувач: strelbitski [Стрельбицький М.Ю.] Час сеансу: 0:02:54 Всього записів: 44 Виділених записів: 1

Рисунок 3.2 – Головне вікно в режимі бази відправлень

3.2.3 Модуль "База надходжень"

Модуль "База надходжень" використовується для обліку пристроїв, які надходять на обслуговування. Головне вікно в цьому режимі показано на рис. 3.3. Залежно від статусу оплати, етапів виконання робіт та встановлення міток, змінюється окреслення шрифту та колір фону відповідних клітинок.

База даних GREMPIS :: База надходжень

Файл Режим Поля Шрифт

Область Місто Від кого

Дата надходження: за останні 30 днів Клас Тип Номер Рік вип. Причина Оплата Номер рахунку Показувати готові прилади

+	Дата надход.	Дата відправки	Клас пристрою	Тип	Номер	Рік вип.	Причина	Область	Місто	Відправник	Док.	Рахунок	Версія Діап.	Плата	Пам.	X	Проц.
	18.01.14		Універсал	02	100001	2014	Повірка	Вінницька	Вінниця	Петренко І.О.	✓		12.01				
	24.01.14	03.02.14	Універсал	02	100004	2014	Ремонт-повір	Вінницька	Бар	Максимович О.В.	✓	162165	12.01				
✓	24.01.14		Універсал	01	100003	2014	Повірка	Вінницька	Бар	Максимович О.В.			12.01				
✓	25.01.14	23.03.14	БІ	02-01	11325	2014	Повернення	Черкаська	Черкаси	ТОВ Черкашаночка	✓	2532	6.01				
	23.03.14	23.03.14	Aplisens	01	1234	2014	Гарантія	Кіровоградсь	Кіровоград	Петренко І.О.			12.01	NXP	B	5	mega128
	23.03.14	23.03.14	Aplisens	01	10320	2014	Гарантія	Кіровоградсь	Кіровоград	Петренко І.О.			12.01	NXP	B	5	mega128
	23.03.14	23.03.14	Універсал	01	1082	2014	Повірка	Хмельницька	Баранівка	Максимович О.В.	✓		12.01				
	23.03.14	23.03.14	Універсал	02	394572	2014	Заміна	Вінницька	Козятин	Максимович О.В.	✓	23423341234					

Користувач: strelbitski [Стрельбіцький М.Ю.] Час сеансу: 0:23:10 Всього записів: 8 Виділених записів: 1

Рисунок 3.4 – Головне вікно в режимі бази надходжень

Модуль надходжень надає такий функціонал:

- додавання нових записів;
- створення нових записів на основі існуючих, або клонування параметрів;
- редагування записів;
- видалення записів;
- встановлення переліку ремонтних робіт;
- встановлення бухгалтерських рахунків;
- додавання інформації про прийом та доставку приладів;
- перегляд історії повірок;
- встановлення статусу оплати.

3.2.4 Модуль "База повірок"

Модуль "База повірок" використовується для додавання інформації про повірки приладів. Головне вікно в режимі бази повірок показано на рис. 3.5.

У цьому модулі користувачеві надаються можливості:

- додавання нових записів;
- редагування записів;
- видалення записів;
- перегляд загальної інформації про пристрій.

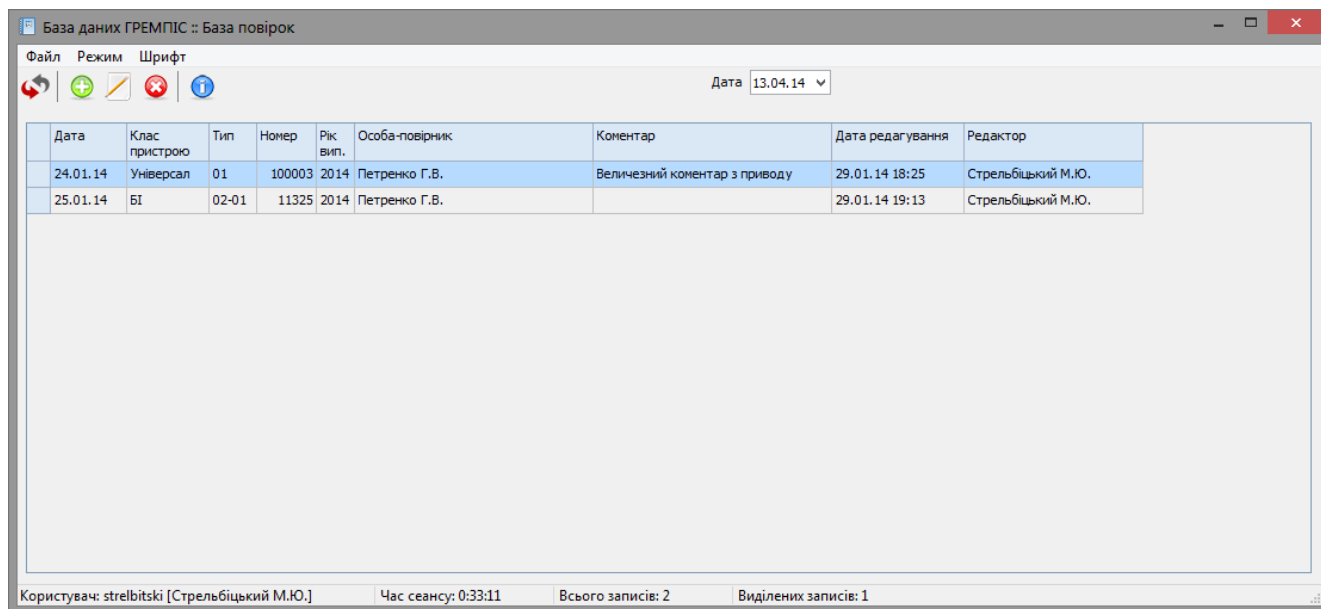


Рисунок 3.5 – Головне вікно в режимі бази повірок

У всіх трьох режимах роботи (бази відправлень, надходжень та повірок) робота із даними виконується за допомогою дочірніх модальних форм, які викликаються за допомогою подій `OnClick()` на кнопках панелі інструментів, або при натисканні «гарячих клавіш» у обробнику подій `ActionList`.

3.2.5 Інші модулі

Модуль фільтрації даних представлений у вигляді панелі інструментів у верхній частині вікна (рис. 3.6). Доступний у режимах баз відправлень та надходжень.

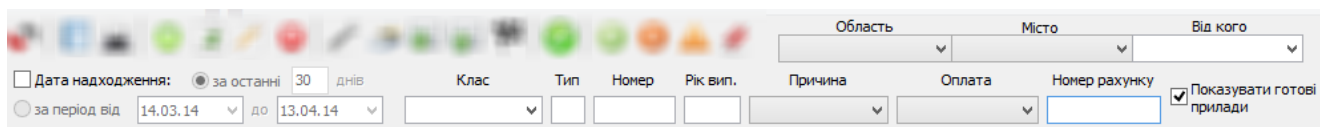


Рисунок 3.6 – Панель інструментів для фільтрації записів

Залежно від обраного режиму, дозволяє фільтрувати записи за ключовими параметрами. При зміні даних у будь-якому із полів фільтрації викликається функція `FilterChange()`, яка зчитує нові параметри з усіх полів та формує новий фільтр, який передається у вигляді параметра властивості `Filter` головної таблиці. У результаті цього відсіюються поля, які не відповідають запиту користувача.

Модуль друку використовується для виведення інформації на принтер або для збереження у електронному форматі. Доступний у режимах баз відправлень та надходжень. У обох режимах надає можливість друку загального, а в режимі бази надходжень також надає можливість друку технологічних карт та бирок для приладів, які перебувають на обслуговуванні. Друк звітів реалізовано за допомогою компонента `FastReport`.

Звіт у `FastReport` та головна таблиця використовують одне джерело даних (`DataSet`), яке взаємодіє із сервером. Таким чином, при друці звітів при попередньому перегляді відображаються дані, які підставляються із `DataSet` до таблиці – тобто, ті ж дані, які відображаються в ній. При друці бирок використовуються лише виділені користувачем записи.

Залежно від обраного користувачем діапазону даних, при друці бирок обираються різні типи шаблонів (для пристроїв «Універсал», для бар'єрів іскрозахисту або загальні). Програма перевіряє діапазон, обирає необхідний та відкриває один із трьох варіантів шаблонів звітів.

Модуль шрифтів є загальним для всіх режимів баз та дозволяє змінювати розмір шрифту в таблиці. Корисний для застосування із використанням моніторів з великим розширенням екрану (рис. 3.7).

	+	Дата надход.	Дата відправки	Клас пристрою	Тип	Номер	Рік вип.	Причина	О
	☐	18.01.14		Універсал	02	100001	2014	Повірка	Ві
	☐	24.01.14	03.02.14	Універсал	02	100004	2014	Ремонт-повір	Ві

Рисунок 3.7 – Використання збільшеного розміру шрифту

3.3 Тестування програмного продукту

Тестування програмного забезпечення – це процес технічного дослідження, призначений для виявлення інформації про якість продукту відносно контексту, в якому він має використовуватись.

Тестування визначається як процес перевірки правильності програми в динаміці її виконання за тестовими даними. При тестуванні виявляються недоліки: відмови і дефекти як причини порушення роботи програми, збої як небажані ситуації, помилки як наслідки збоїв та ін. Тестування вважається успішним, якщо знайдено дефект або помилка, і вони відразу усуваються. Зазвичай для проведення тестування застосовуються методи структурного ("білий ящик") та функціонального ("чорний ящик") тестування [16].

При функціональному тестуванні вихідний код недоступний. Суть полягає в перевірці відповідності поведінки програми її зовнішній специфікації. Критерієм повноти тестування вважається перебір всіх можливих значень вхідних даних, що здійснити на практиці надзвичайно важко.

При структурному тестуванні текст програми відкритий для аналізу. Суть даного методу полягає в перевірці внутрішньої логіки програмного забезпечення.

Повним тестуванням у цьому випадку буде таке, що приведе до перебору всіх можливих шляхів на графі передач керування програми. Число таких шляхів може досягати десятків тисяч. Крім того, виникає питання про створення тестів, що забезпечують дане покриття. Здійснити повне всеохоплююче тестування навіть простої програми вкрай важко, а часом і неможливо в силу обмеженості часу й ресурсів. Отже, необхідно мати певні критерії, за якими мають обиратися контрольні приклади та критерії зупинки процесу тестування.

Оскільки метод "білого ящика" є досить громіздким, а програма досить функціональна, доречно буде скористатися методом "чорного ящика", не втручаючись у вихідний текст програми.

Для тестування працездатності роботи програми створимо тестовий пристрій з усіма параметрами, додамо його в бази надходжень та перевірок, виставимо перелік ремонтних робіт та рахунків, а також перевіримо роботу фільтрів та друк звітів.

При першому запуску програми з'являється вікно авторизації (рис. 3.8).

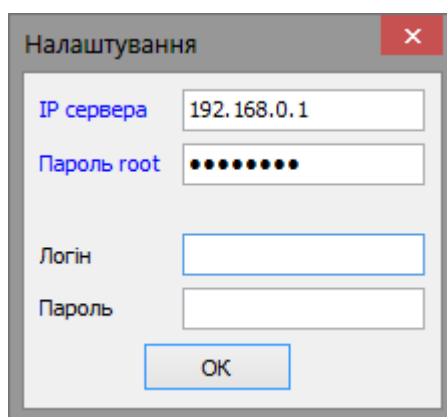


Рисунок 3.8 – Вікно авторизації

Створимо новий пристрій за заданими параметрами (рис. 3.9). Після додавання нового запису він відмічається у таблиці синім кольором (рис. 3.10). Перейдемо у режим бази надходжень та додамо цей пристрій як такий, що надійшов до ремонту (рис. 3.11).

Новий пристрій

Номер Тип Дата відправки

Рік випуску Версія Область

Плата Частота Місто

Пам'ять X Процесор Перший встан.

☒ Додати коментар Поточний встан.

Коментар

☒ RS232 ☐ DTR Час роботи акумулятора: ☐ + ☐ -

☐ RS485_1 ☒ Modem Час прогону:

☒ RS485_2

Примітка

Записів додано: 0

Рисунок 3.9 – Додавання нового пристрою

1411	02							2014	08.03.14	Вінницька	Коз
1897	01	12.12	NXP	mega128	B	4	7,37280	2014	13.04.14	Хмельницька	Хме
100001	02	12.01						2014		Вінницька	Він

Рисунок 3.10 – Новий запис у базі відправлень

Новий запис

Пошук по базі

Номер ☐ Інший пристрій із таким номером

	Номер	Клас	Тип	Рік вип.	Версія	Плата	Частота	Довжина	Встановив
▶	1897	Універсал	01	2014	12.12	NXP	7,37280		Хмельницький-Енерго

Клас Тип Дата отримання Документи ☒

Рік випуску Версія / діапазон Причина

Плата Частота / довжина Область

Пам'ять X Процесор Місто

Коментар

Від кого

Спосіб одерж.

Номер декларації або П.І.Б.

Записів додано: 0

Рисунок 3.11 – Додавання пристрою в базу надходжень

Додамо для пристрою набір робіт (рис. 3.12). По завершенні цієї процедури бухгалтер зможе виставити рахунок. По завершенні процедури рядок виділяється напівжирним окресленням тексту.

Код	Вид робіт	Ціна	К-ть	Сума
16	Повірка	250,00	1	250,00
14	Діагностика роз'євів	164,50	3	493,50
Загальна вартість (з ПДВ): 743,50				

Рисунок 3.12 – Додавання переліку ремонтних робіт

Наступним етапом є виставлення рахунку (рис. 3.13). По його завершенні текст у колонці "Причина надходження" виділяється напівжирним та курсивом.

Виставити рахунок

Бухгалтерські дані
 Номер рахунку
 5241 від 13.04.2014
 Видал. дані
 Виставити

Інформація про ремонт
 Перелік кодів ремонтних робіт:
 [16] 3x[14]
 Загальна сума: **743,50 грн.**
 Запис додано користувачем:
Стрельбіцький М.Ю.
 13.04.2014 22:35:22

Рисунок 3.13 – Виставлення рахунку

В переліку ремонтних робіт також зазначався пункт "Повірка". Переходимо у режим бази повірок та додаємо відповідний запис (рис. 3.14).

Нова повірка

Номер 1897

	Дата	Причина	Номер	Клас	Тип	Рік
▶	13.04.14	Ремонт	1897	Універсал	01	2014

Дата 13.04.2014 Повірник Петренко Г.В.

Коментар

Пристрій в нормі.

Додано записів: 0

Додати Завершити

Рисунок 3.14 – Додавання пристрою до бази повірок

По завершенні процедури оплати можна встановити статус "Оплачено" та відправити пристрій (рис. 3.15).

Дані про доставку клієнту

Дата 13.04.2014 Спосіб доставки # Самовивіз

Номер декларації або П.І.Б.

Петришин І.О.

Востаннє змінено:

ОК

Рисунок 3.15 – Інформація про відправлення пристрою

В кінцевому результаті рядок з пристроєм набуде вигляду, як показано на рис. 3.16. В першому стовпці з'явиться позначка, яка свідчить про те, що повірку пристрою було здійснено і про це є запис у базі повірок. Пристрій також можна відмітити як готовий, або архівний. Після цього фон рядку зміниться на білий.

Дата надх.	Дата відпр.	Клас	Тип	Номер	Рік вип.	Причина надх.	Область	Місто	Відправник	Док.	Рахунок	Версія Діп.	Плата
13.04.14	13.04.14	Універсал	01	1897	2014	Ремонт	Хмельницька	Хмельницький	Хмельницький-Енерго	✓	5241 від 13.04.2014	12.1	

Рисунок 3.16 – Вигляд рядка по завершенні всіх операцій

Перевіримо роботу фільтрів. Для цього знімемо позначку з пункту "Показувати готові прилади" та оберемо лише прилади класу "Універсал" (рис. 3.17).

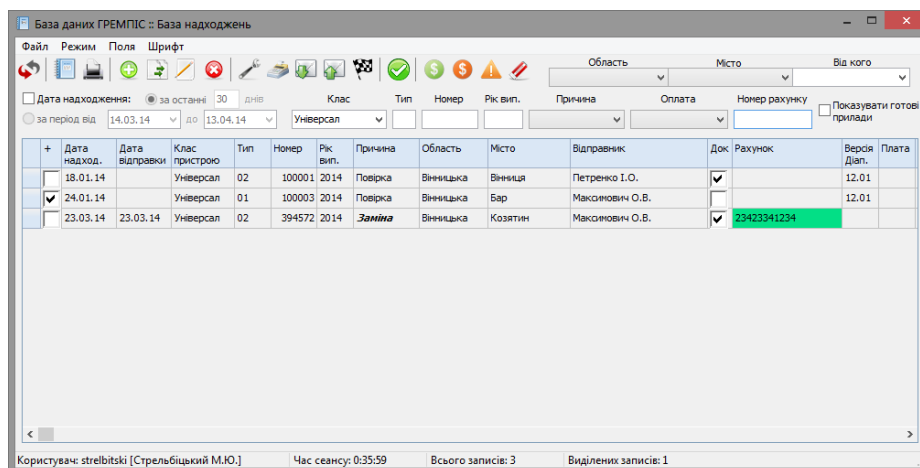


Рисунок 3.17 – Демонстрація роботи фільтрів

Перевіримо друк звітів. На загальний друк виводяться всі видимі рядки (рис. 3.18).

Дата надх.	Дата відпр.	Клас	Тип	Номер	Рік вип.	Причина надх.	Область	Місто	
18.01.14		Універсал	02	100001	2014	Повірка	Вінницька	Вінниця	Петр
24.01.14		Універсал	01	100003	2014	Повірка	Вінницька	Бар	Мак
23.03.14	23.03.14	Універсал	02	394572	2014	Заміна	Вінницька	Козятин	Мак

Рисунок 3.18 – Друк загального звіту

Для друку технологічних карт виділимо необхідні прилади і спробуємо вивести їх на друк (рис. 3.19).

Необхідно перевірити наявність файлу libmysql.dll в папці з програмою. Наявність файлу interface.ini необов'язкова. Для запуску програми потрібно запустити на виконання файл GrempisDB.exe. При першому запуску з'явиться вікно авторизації (див. рис. 3.8). Якщо обліковий запис не створено, програма запропонує це зробити (рис. 3.20).

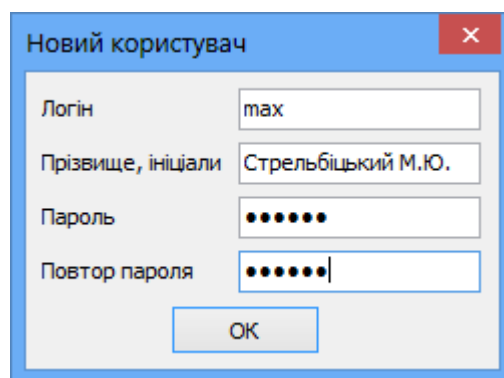


Рисунок 3.20 – Створення нового користувача

Після створення облікового запису з'явиться головне вікно програми (див. рис. 3.2). Для роботи із базою даних можна використовувати кнопки панелі інструментів або гарячі клавіші:

- У всіх режимах (відмічені зірочкою функції недоступні в режимі бази перевірок):



Оновити сторінку – для випадків проблем зі з'єднанням;



Друк звіту (Ctrl + P)* – друк вмісту таблиці або збереження на носії;



Додати запис (Insert) – додає новий запис до бази даних;



Клонувати запис* – додає запис на основі даних виділеного;



Редагувати запис – змінює інформацію у виділеному записі;



Видалити запис (Del) – видаляє обрані записи;



«Увага»* – відмічає пристрій іншим кольором на фоні решти.



Видалити мітку* - видаляє відмітку кольором.

- У режимі бази надходжень:



Історія ремонту – відображає історію обраного пристрою з коментарями;

- У режимі бази відправлень:



Друк технологічних карт (Ctrl + T) – для обраного діапазону пристроїв;



Перелік кодів робіт – введення переліку проведених робіт та їх оцінка;



Виставлення рахунку (F4) – введення бухгалтерської інформації;



Інформація про прийом (F5) – введення інформації про відправника;



Інформація про доставку (F3) – введення інформації про доставку;



Відмітка про завершення – прилад готовий;



Історія перевірок – відображає історію перевірок обраного пристрою;



«Оплачено» – відмічає пристрій, по якому здійснено оплату;



«Боржник» – відмічає пристрій, по якому в клієнтів є борги;

- У режимі бази перевірок:



Інформація – детальна інформація по виділеному пристрою.

Для того, щоб додати, клонувати або редагувати пристрій, натисніть відповідну кнопку і заповніть поля у формі (див. рис. 3.9). Після цього натисніть кнопку «Додати». У разі успіху лічильник доданих записів біля кнопки збільшиться на одиницю.

Щоб ввести всі ремонтні дані по приладу, необхідно виконати такі дії:

1. Додати прилад, який уже є в базі відправлень, або створити новий.
2. Ввести перелік кодів ремонтних робіт. Після цього у таблиці стрічка відображатиметься напівжирним накресленням шрифту.
3. Виставити рахунок за ремонт. Накреслення шрифту змінюється на напівжирний курсив.
4. Відмітити прилад як оплачений при отриманні коштів від клієнта.
5. Ввести дані про відправлення приладу.
6. Відмітити прилад як готовий.

Для друку технологічних карт необхідно вибрати лише ті прилади, для яких вони друкуватимуться. Залежно від діапазону, який обирається, програма підбирає відповідний шаблон звіту. При друці звіту на принтер виводиться вміст таблиці.

У разі виникнення помилок програма виводить відповідні повідомлення на екран.

3.5 Висновки

У третьому розділі описані процеси розробки та тестування роботи програми, а принципи роботи в програмі описані в керівництві користувача.

Для розробки було обрано мову програмування C++. До її переваг можна віднести швидкодію та наявність інструментів для роботи із MySQL-сервером.

У ході тестування було виявлено, що програма працює коректно та відповідає поставленим до неї вимогам.

Для користування програмою можна скористатись керівництвом користувача, яке коротко описує, як вона працює.

4 ОХОРОНА ПРАЦІ

4.1 Характеристика професії та приміщення

Характеристику професії «Інженер з програмного забезпечення комп'ютерів» наведено в таблиці 4.1 [17-19].

Таблиця 4.1 – Загальні характеристики професії «Інженер з програмного забезпечення комп'ютерів»

Назва	Опис
1	2
Розділ	2. Професіонали
Галузі застосування знань	Цей розділ вміщує професії, що передбачають високий рівень знань у галузі фізичних, математичних, технічних, біологічних, агрономічних, медичних чи гуманітарних наук. Професійні завдання полягають у збільшенні існуючого фонду (обсягу) знань, застосуванні певних концепцій, теорій та методів для розв'язання певних проблем чи в систематизованому викладенні відповідних дисциплін у повному обсязі. До цього розділу належать професії, що вимагають від працівника (з урахуванням кола та складності певних професійних завдань та обов'язків) кваліфікації за: 1) дипломом про повну вищу освіту, що відповідає рівню спеціаліста, магістра; 2) дипломом про присудження наукового ступеня: - кандидата наук; - доктора наук; 3) атестатом про затвердження вченого звання: - старшого наукового співробітника; - доцента; - професора.
Кваліфікаційні вимоги	Провідний інженер з програмного забезпечення комп'ютерів: повна вища освіта відповідного напрямку підготовки (магістр, спеціаліст). Стаж роботи за професією інженера з програмного забезпечення комп'ютерів I категорії — не менше 2 років. Інженер з програмного забезпечення комп'ютерів I категорії: повна вища освіта відповідного напрямку підготовки (магістр, спеціаліст); для магістра — без вимог до стажу роботи,

Продовження таблиці 4.1

Кваліфікаційні вимоги	спеціаліста — стаж роботи за професією інженера з програмного забезпечення комп'ютерів виробництвом II категорії — не менше 2 років. Інженер з програмного забезпечення комп'ютерів II категорії: повна вища освіта відповідного напрямку підготовки (спеціаліст). Стаж роботи за професією інженера з програмного забезпечення комп'ютерів — не менше 1 року. Інженер з програмного забезпечення комп'ютерів: повна вища освіта відповідного напрямку підготовки (спеціаліст) без вимог до стажу роботи.
Код професії	2131.2
Код за ДКХПП	1
Завдання та обов'язки в сфері охорони праці	- перед початком роботи проводити огляд свого робочого місця, звертаючи увагу на справність ЕОМ, наявність та справність огорожень, захисних пристроїв, блокувальних і сигнальних пристроїв, засобів індивідуального і комплексного захисту, засобів пожежогасіння, заземлення, чистоту обладнання, стан проходів, майданчиків, проїздів, сходових пристосувань, а також присутність сміття, мотлоху; - при виявленні недоліків, доповідати про це керівнику дільниці і по його вказівці приймає участь у їх усуненні; - використовувати засоби захисту і пристрої, які забезпечують безпеку праці. Дотримуватись всіх вимог інструкцій з охорони праці; - повідомляти керівнику про кожний випадок травми, отруєння, опіку, які отримані особисто або іншими робітниками, а також про загорання або про іншу аварійну ситуацію; - знати і виконувати вимоги інструкцій про охорону праці, правила поведінки з машинами, механізмами, устаткуванням та іншими засобами виробництва, користуватися засобами колективного та індивідуального захисту; - дотримуватися зобов'язань щодо охорони праці, передбачених колективним договором та правилами внутрішнього розпорядку; - проходити у встановленому порядку попередні та періодичні медичні огляди; - особисто вживати посильні заходи щодо усунення будь-якої виробничої ситуації, що створює загрозу його життю чи здоров'ю, повідомляти про небезпеку свого безпосереднього керівника; - знати прийоми надання першої медичної допомоги тощо.

Об'єктом дослідження є відділ програмного забезпечення у м. Вінниці.

Умовно приймаємо, що приміщення відділу знаходиться на другому поверсі адміністративної будівлі. Загальна площа становить $6 \cdot 5 = 30 \text{ м}^2$, висота – 2,5 м, приміщення має два вікна. Кількість працюючих у приміщенні – 7 осіб. Отже, на одного працюючого в приміщенні припадає $30 : 7 = 4,3 \text{ (м}^2\text{/ос.)}$ робочої площі. Згідно із СНиП 2.09.04-87 [17], на кожного працюючого в управлінських приміщеннях повинно припадати не менше $4 \text{ (м}^2\text{/ос.)}$ робочої площі. Відповідно до НПАОП 0.00-1.28-10 [18], на одного працівника, який застосовує ЕОМ, має припадати не менше 6 м^2 , а об'єм повітря в приміщенні має бути не менше 20 м^3 . Висота приміщення – не менше 2,5 м. Отже, нормативи розмірів та забезпечення працюючих робочою площею та об'ємом повітря у відділі програмного забезпечення не дотримано. Передбачено два вікна (розмірами довжина 1,4 м., висота 1,5 м.).

У приміщенні розташовано 7 комп'ютерів. Напруга джерела живлення комп'ютерів у приміщенні – 220 В. У приміщенні розміщені 7 письмових столів, одна шафа для зберігання документів, один холодильник.

За небезпекою ураження електричним струмом, згідно із ПУЕ:2009 [19], приміщення відділу програмного забезпечення належить до приміщень без підвищеної небезпеки ураження електричним струмом працюючих.

Відповідно до типових норм належності вогнегасників НАПБ Б.03.001–2004 [20] для нашого приміщення доцільно обрати два вуглекислотних вогнегасника (ВВК–5), а тип пожежної сигналізації димова або полум'я.

Схема приміщення наведена на рисунку 4.1.

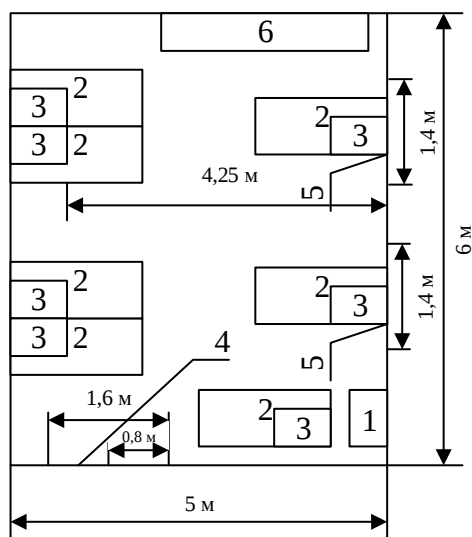


Рисунок 4.1 – Схема приміщення

Умовні позначення:

- 1 – холодильник;
- 2 – робочі місця (столи);
- 3 – персональні комп'ютери;
- 4 – дверний отвір;
- 5 – вікна;
- 6 – шафа.

4.2 Аналіз стану охорони праці в приміщенні

Аналіз метеорологічних умов проводиться для категорії робіт легка 1б у теплий період року на непостійному робочому місці. Результати досліджень параметрів мікроклімату та їх співставлення із оптимальними та допустимими нормами наведено у таблиці 4.2. Оптимальні значення також відповідають нормам НПАОП 0.00-1.28-10 [18].

Таблиця 4.2 – Аналіз метеорологічних умов на робочому місці

№	Параметр мікроклімату	Оптимальне значення	Допустимі значення	Фактичне значення
1	Температура, °C	22-24	19-30	16
2	Відносна вологість, %, не більше	40-60	75	36
3	Швидкість руху повітря, м/с, не більше	0,2	0,1-0,3	0,12

Згідно із отриманими даними, можна зробити висновок, що температура повітря в приміщенні (16 °C) не відповідає допустимим нормам (19-30 °C), відносна вологість (36%) також не відповідає нормам (40-60%), а швидкість руху повітря (0,12 м/с) є допустимою, але неоптимальною (0,2 м/с). Тому, оскільки два із трьох показників є недопустимими, приміщення має несприятливі умови праці.

Чинними в Україні є значення ГДК, що містяться у переліку СНиП 4617-88 «Предельно допустимые концентрации вредных веществ в рабочей зоне» [21]. Порівняння фактичних параметрів із нормативними наведено в таблиці 4.3.

Таблиця 4.3 – Аналіз хімічного складу повітря робочої зони

№	Речовина	Клас небезпеки	ГДК, мг/м ³	Фактична концентрація, мг/м ³	Відносна концентрація
1	Оксид вуглецю	IV	20	20	1,0
2	Озон	I	0,1	0,16	1,6
3	Оксид азоту	III	2	4,5	2,25
4	Формальдегід	II	0,5	0,1	0,2
Всього:					5,05 > 1

Згідно із даними таблиці, у приміщенні спостерігається перенасичення озоном (0,16 мг/м³ при ГДК 0,1 мг/м³), оксидом азоту (4,5 мг/м³ при ГДК 2 мг/м³) та гранична насиченість повітря оксидом вуглецю (20 мг/м³). При цьому сума відношень фактичних концентрацій речовин у приміщенні до їх ГДК перевищує 1. Це свідчить про те, що загалом повітря у робочій зоні створює несприятливі для роботи умови праці.

Для аналізу достатності природного освітлення наведемо схему приміщення розрахунку (рис. 4.2).

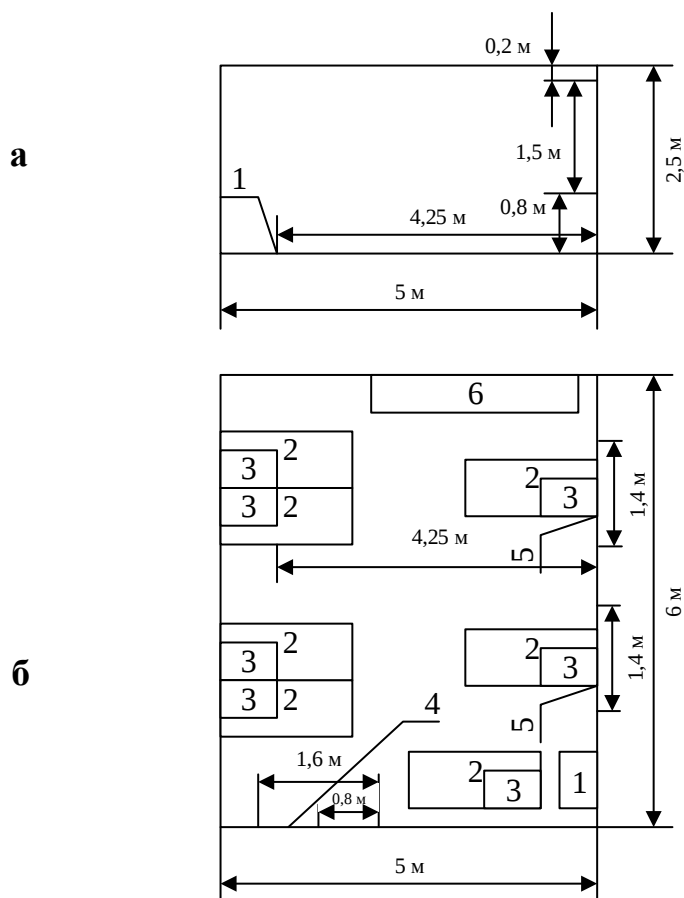


Рисунок 4.2 – Схема розрахунку природного освітлення

де: а, б – вигляд збоку та зверху.

Порівнюючи значення нормованого коефіцієнта природного освітлення і фактичного освітлення, можна зробити висновок про необхідність додаткових заходів щодо поліпшення природного освітлення у досліджуваному приміщенні. Ефективність додаткових заходів щодо поліпшення природного освітлення необхідно перевірити відповідними розрахунками.

Нормоване значення КПО визначаємо з ДБН В.2.5–28–2006 «Природне і штучне освітлення» [22] (приймаємо роботу середньої точності). Орієнтацію світлових прорізів за сторонами горизонту приймаємо такою, що вікна виходять на північний схід. Тоді КПО буде становити 1,5, а коефіцієнт світлового клімату (відповідно до табличного значення) буде становити 0,9. Тоді нормоване значення КПО складе:

$$e_N = 1,5 \cdot 0,9 = 1,35.$$

Для робочих місць, де застосовують ЕОМ, значення КПО має становити не менше 1,5%. Для визначення достатності природного освітлення потрібно розрахувати фактичне значення КПО за формулою:

$$e_\phi = \frac{100 \cdot S_B \cdot \tau_3 \cdot r_1}{S_{\Pi} \cdot \eta_B \cdot K_3 \cdot K_{\text{буд}}} \quad (4.1)$$

Площа всіх вікон у приміщенні становить $S_B = 1,4 \cdot 1,5 \cdot 2 = 4,2 \text{ м}^2$.

Площа підлоги приміщення $S_{\Pi} = 6 \cdot 5 = 30 \text{ м}^2$.

Загальний коефіцієнт світлопроникності віконного прорізу $\tau_3 = 0,65 \cdot 0,9 \cdot 1 \cdot 1 \cdot 1 = 0,585$.

Світлова характеристика вікна η_B розраховується за даними таблиці:

Відношення довжини приміщення L до його глибини $B = 6/5 = 1,2$.

Відношення глибини приміщення B до висоти від рівня робочої поверхні до верхнього краю вікна $h = 5/1,5 = 3,33$.

Таким чином, $\eta_B = 16,5$.

Коефіцієнт, що враховує затемнення вікон іншими будинками розраховується за даними таблиці. Оскільки $P/H_{\text{буд}} = 1$, то $K_{\text{буд}} = 1,4$.

Приймаємо коефіцієнт запасу 1,3. Тоді $K_3 = 1,3$.

Для розрахунку коефіцієнта r_1 необхідно розрахувати такі параметри:

- 1) середньозважений коефіцієнт відбиття;
- 2) відношення глибини приміщення до висоти від рівня умовної робочої поверхні до верху вікна;
- 3) відношення відстані до розрахункової точки від зовнішньої стіни до глибини приміщення;
- 4) відношення довжини приміщення до його глибини.

Розрахуємо середньозважений коефіцієнт відбиття $\rho_{\text{сер}}$ за формулою:

$$\rho_{\text{сер}} = \frac{\rho_{\text{стелі}} \cdot S_{\text{стелі}} + \rho_{\text{стін}} \cdot S_{\text{стін}} + \rho_{\text{підлоги}} \cdot S_{\text{підлоги}}}{S_{\text{стелі}} + S_{\text{стін}} + S_{\text{підлоги}}}. \quad (4.2)$$

Так, приймаємо, що стеля свіжопобілена, тоді $\rho_{\text{стелі}} = 0,8$; приймаємо, що стіни обклеєні темними шпалерами, тоді $\rho_{\text{стін}} = 0,15$; приймаємо, що підлога пофарбована в оливково-зелений колір, тоді $\rho_{\text{підлоги}} = 0,2$. Отже, за формулою (4.2), середньозважений коефіцієнт відбиття рівний:

$$\rho_{\text{сер}} = \frac{0,8 \cdot 30 + 0,15 \cdot 55 + 0,2 \cdot 30}{30 + 55 + 30} = 0,333.$$

Таким чином, $\rho_{\text{сер}} = 0,333$.

Порахувавши значення параметрів, що характеризують адміністративне приміщення (відділ програмного забезпечення), визначаємо коефіцієнт r_1 .

- відношення глибини приміщення В до висоти від рівня умовної робочої поверхні до верху вікна h: $5/1,5 = 3,33$;
- відношення відстані до розрахункової точки від зовнішньої стіни 1 до глибини приміщення В: $4,25/5 = 0,85$;
- відношення довжини приміщення L до його глибини В: $6/5 = 1,2$.

Враховуючи наведені вище співвідношення та табличні дані, коефіцієнт, який враховує відбиття світла від внутрішніх поверхонь приміщення, буде дорівнювати $r_1 = 1,85$.

Отже, фактичне значення коефіцієнта природного освітлення e_{ϕ} , за формулою (4.1), дорівнює:

$$e_{\phi} = \frac{100 \cdot 4,2 \cdot 0,585 \cdot 1,85}{30 \cdot 16,5 \cdot 1,3 \cdot 1,4} = 0,50.$$

Таким чином, фактичне значення природного освітлення менше нормованого ($0,50 < 1,35 (1,5)$).

Приймаємо, що для освітлення відділу програмного забезпечення застосовуються люмінесцентні лампи потужністю 60 Вт. Система освітлення – загальна. Отже, нормоване значення освітленості повинне становити не менше 300 люкс (ДБН В. 2.5–28–2006) [22]. Схема розміщення світильників наведено на рисунку 4.3.

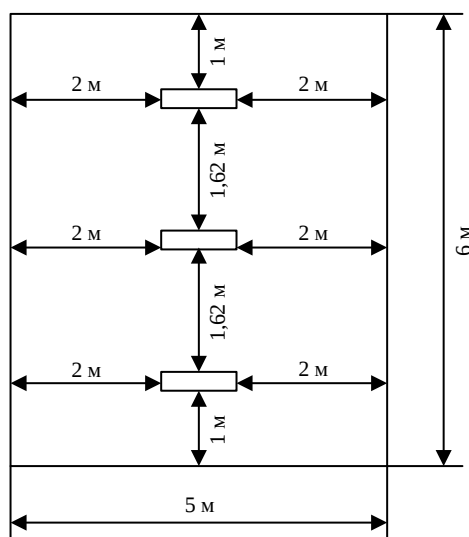


Рисунок 4.3 – Схема розміщення світильників

Розрахуємо фактичне освітлення E_{ϕ} . Приймаємо, що потужність ламп – 60 Вт, а кількість ламп у світильнику – 2 шт. Фактичне значення штучного освітлення розраховуємо за формулою:

$$E_{\phi} = \frac{F_{\text{л}} \cdot \eta_{\text{в}} \cdot N \cdot n}{S \cdot k \cdot z}, \quad (4.3)$$

де $F_{\text{л}}$ – світловий потік однієї лампи, лм;

$\eta_{\text{в}}$ – коефіцієнт використання світлового потоку;

N – кількість світильників у приміщенні, яке досліджується, шт.;

n – кількість ламп у кожному світильнику, шт.;

S – площа приміщення, м^2 ;

k – коефіцієнт запасу;

z – коефіцієнт нерівномірності освітлення.

Оскільки світловий потік люмінесцентної лампи ЛДУ 40 рівний 2100 лм, а лампи ЛДУ 80 – 3740 лм, то вважатимемо, що світловий потік для люмінесцентної лампи потужністю 60 Вт складатиме $(2100+3740)/2 = 2920$ лм.

Коефіцієнт використання світлового потоку η_v – приймаємо середнє значення: $(0,4+0,6)/2 = 0,5$.

Кількість світильників, за умовою, $N = 3$.

Кількість ламп у світильнику, за умовою, $n = 2$.

Площа приміщення $S = 30$ м².

Коефіцієнт запасу, за таблицею, $k = 1,5$.

Коефіцієнт нерівномірності освітлення для люмінесцентних ламп $z = 1,1$.

Таким чином, фактичне значення штучного освітлення, за формулою (4.3), буде рівним:

$$E_{\phi} = \frac{2920 \cdot 0,5 \cdot 3 \cdot 2}{30 \cdot 1,5 \cdot 1,1} = 177,0 \text{ лк.}$$

Отже, фактичне значення штучного освітлення менше нормованого ($177 \text{ лк} < 300 \text{ лк}$), а це свідчить про недостатність штучного освітлення в приміщенні.

4.3 Розробка заходів щодо покращення умов праці на робочому місці

В ході виявлення недоліків під час проведення розрахунків щодо аналізу загальної характеристики дотримання стану охорони праці в приміщенні, умов праці, норм природного і штучного освітлення отримані результати (недоліки та заходи щодо їх усунення) заносимо в таблицю 4.4.

Таблиця 4.4 – Розробка заходів покращення стану охорони та умов праці

№ з/п	Виявлені недоліки	Заходи покращення умов праці
1	Знижена температура повітря	- виконати утеплення стін та вікон, щоб тепле повітря утримувалось в приміщенні; - вдосконалити систему опалення; - встановити кондиціонер, тощо.
2	Низька відносна вологість повітря	- частіше провітрювати приміщення; - використовувати в приміщенні вази та інші ємності, з яких може випаровуватись вода; - встановити кондиціонер, тощо.
3	Підвищена концентрація в повітрі озону та оксиду азоту	- запобігти проникненню шкідливих речовин у повітря робочої зони за рахунок герметизації; - частіше провітрювати приміщення; - вдосконалити систему вентиляції; - встановити кондиціонер, тощо.
4	Недостатній рівень природного освітлення	- застосувати матеріали, що підвищують відбиття світла від внутрішніх поверхонь приміщення; - зняти сонцезахисні жалюзі, штори, захисні козирки тощо; - збільшити коефіцієнт запасу шляхом застосування скла, яке краще пропускає природне освітлення; - застосувати штучне освітлення, тощо.
5	Недостатній рівень штучного освітлення	- замінити лампи на інші, більшої потужності; - збільшити кількість ламп у світильниках; - збільшити кількість світильників; - використовувати комбіноване освітлення, тощо.

4.4 Висновки

Під час виконання четвертого розділу було проведено ряд розрахунків щодо виконання вимог охорони праці в приміщенні відділу програмного забезпечення. Дотримання цих вимог є важливим для збереження працездатності та здоров'я працівників.

У першому пункті було наведено характеристику професії «Інженер з програмного забезпечення комп'ютерів». Згідно із Національним класифікатором

професій, ця професія належить до розділу «Професіонали». Було наведено також і обов'язки працівника в сфері охорони праці. Також було описано приміщення відділу програмного забезпечення у м. Вінниці. Зокрема, було наведено основні параметри приміщення та його схему. Було встановлено, що на одного працівника припадає 4,28 м² робочої площі та 10,7 м³ повітря. Ці дані свідчать про невідповідність нормам охорони праці при експлуатації електронно-обчислювальних машин.

В другому пункті було проаналізовано стан охорони праці у відділенні програмного забезпечення. Зокрема, при аналізі мікроклімату було встановлено, що нормам не відповідає температура повітря у приміщенні та вологість повітря. В ході аналізу хімічного стану повітря було визначено, що концентрація у повітрі робочої зони озону та оксиду азоту підвищена. При оцінці ефективності природного освітлення було виявлено, що його рівень недостатній (значення КПО $0,5 < 1,35$ (1,5)). Рівень штучного освітлення також виявився недостатнім ($177 < 300$ лк).

У третьому пункті було запропоновано заходи щодо поліпшення умов праці. Зокрема, було запропоновано виконати утеплення стін та вікон, вдосконалити систему опалення, встановити кондиціонер, поліпшити герметичність приміщення, замінити лампи на більш потужні, тощо.