Міністерство освіти і науки України

Одеський національний політехнічний університет

Інститут комп’ютерних систем  
Кафедра комп’ютерних інтелектуальних систем і мереж

**КУРСОВА РОБОТА**

з дисципліни «Системне програмування»

за темою

«Розробка програмного забезпечення «Основні засоби»

Частина 3. Керування сутностями: внутрішня

накладна, документ

Виконала:

студентка 3-го курсу

групи АМ-182

Хора В.П.

Перевірив:

Блажко О. А.

Одеса-2020

**Анотація**

В курсовій роботі розглядається процес розробки програмного забезпечення «Основні засоби Частина 3. Керування сутностями: внутрішня

накладна, документ інформаційної системи бухгалтерії. Робота виконувалась в команді з чотирьох учасників, тому в пояснювальній записці представлено процес розробки лише однієї частини програмного забезпечення – «Документи». В пояснювальній записці представлено опис:

* старого програмного забезпечення, розробленого з використанням мови DelphiObjectPascal та системи керування базами даних FireBird;
* оновлену семантичну модель структур даних;
* нову реляційну модель бази даних з урахуванням частини таблиць;
* збережені процедури на мові FireBirdStoredProcedure з урахуванням частини таблиць;
* екранних форм нового програмного забезпечення у середовищі VisualStudio;
* програних класів на мові C#.

Зміст роботу буде цікавим для спеціалістів з розробки корпоративних інформаційних систем.

**Перелік скорочень**

ОС – операційна система

ІС – інформаційна система

БД – база даних

СКБД – система керування базами даних

ПЗ – програмне забезпечення

UML – уніфікована мова моделювання

**Зміст**

|  |  |  |
| --- | --- | --- |
|  | Завдання до курсової роботи | 5 |
| 1. | Моделювання предметної галузі | 8 |
| 1.1. | Аналіз інтерфейсу користувача програмного забезпечення «Основні засоби»інформаційної системи роботи бухгалтерії | 8 |
| 1.1.1. | Призначення програмного забезпечення «Основні засоби» | 8 |
| 1.1.2. | Коротка інструкція користувача з використання програмного забезпечення «Основні засоби» | 8 |
| 1.2. | Моделювання структур даних | 14 |
| 1.2.1. | Опис назв об`єктів та їх атрибутів на основі аналізу інструкцій користувача по роботі з програмним забезпеченням | 14 |
| 1.2.2. | UML-діаграма концептуальних класів | 14 |
| 2. | Розробка бази даних | 15 |
| 2.1. | Проектування реляційної моделі бази | 15 |
| 2.1.1. | Опис реляційної моделі даних | 15 |
| 2.1.2. | Опис типів даних атрибутів таблиць реляційної моделі | 16 |
| 2.2. | Опис процесу інсталяція СКБД FireBird в ОС Linux | 16 |
| 2.2.1. | Встановлення ОС Linux | 16 |
| 2.2.2. | Описати кроки інсталяції СКБД FireBird з прикладами кроків у вигляді скріншотів екранів | 18 |
| 2.2.3. | Описати кроки створення БД в СКБД FireBird з використанням утиліти isql та прикладами кроків у вигляді скріншотів екранів  (назва БД повинна співпадати з вашим прізвищем у транслітерації) | 20 |
| 2.2.4. | Опис кроків налаштування доступу до БД з використанням DBeaver | 21 |
| 2.3. | Конструювання БД в СКБД FireBird | 24 |
| 2.3.1. | Опис SQL-скрипта створення структури БД | 24 |
| 2.3.2. | Опис SQL-скрипта заповнення таблиць БД тестовими даними | 26 |
| 2.4. | Опис пакетів збережених процедур FireBird | 28 |
| 3. | Розробка графічного інтерфейсу доступу до БД | 31 |
| 3.1. | Опис екранних форм в програмному середовищі VisualStudio | 31 |
| 3.2. | Опис програмних класів на мові C# | 34 |
|  | Висновки | 35 |

ЗАВДАННЯ

В таблиці представлено опис етапів виконання завдань курсової роботи першого модульного контролю.

Таблиця – Опис етапів виконання завдань курсової роботи

|  |  |  |
| --- | --- | --- |
| № етапу | Опис етапу | Опис завдання |
| 1 | 2 | 3 |
| 1 | Аналіз інтерфейсу користувача програмного модуля | 1. переглянути роботу програмного модуля в різних режимах функціонування (перегляд даних, пошук даних, додавання, зміна та видалення даних); 2. створити інструкція користувача, яка містить скріншоти екранних форм програмного модуля та коментарі до роботи |
| 2 | Моделювання структур даних | 1. визначити назви об`єктів та їх атрибутів на основі аналізу інструкцій користувача по роботі з програмним модулем; 2. створити UML-діаграму концептуальних класів; 3. визначити назви класів для індивідуального завдання; |
| 3 | Проектування реляційної моделі | 1. створити реляційну модель даних на основі UML-діаграми концептуальних класів; 2. окремо визначити таблиці реляційної моделі класів з індивідуального завдання; 3. визначити типи даних атрибутів таблиць реляційної моделі у формі таблиці: назва таблиці БД, назва атрибута, тип атрибута ( ціле, строкове з розміром, дата); |
| 4 | Інсталяція СКБД FireBird в ОС Linux | 1. інсталювати ОС Linux, наприклад, Ubuntu, створивши власного користувача з правами адміністратора (назва повинна співпадати з вашим прізвищем у транслітерації) 2. описати кроки інсталяції СКБД FireBird з прикладами кроків у вигляді скріншотів екранів 3. описати кроки створення БД в СКБД FireBird з використанням утиліти isql та прикладами кроків у вигляді скріншотів екранів (назва БД повинна співпадати з вашим прізвищем у транслітерації) 4. описати кроки налаштування доступу до БД з використанням менеджера роботи з СКБД різних виробників DBeaver та за кожним кроком представити скріншоти екранів |

Продовження табл. 1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 5 | Конструювання БД в СКБД FireBird | 1. на основі реляційної моделі БД створити скрипт з SQL-командами CREATETABLEта ALTERTABLEдля таблиць індивідуального завдання 2. завантажити скрипт в БД |
| 6 | Заповнення таблиць БД тестовими даними | 1. створити скрипт із прикладами SQL-команд внесення даних до таблиць БД з індивідуального завдання (10 INSERT-команд) 2. створити скрипт із прикладами SQL-команд зміни даних в таблицях БД з індивідуального завдання  (3 UPDATE-команди) 3. створити скрипт із прикладами SQL-команд видалення даних з таблиць БД з індивідуального завдання (3 DELETE-команди) |
| 7 | Створення пакетів збережених процедур FireBird | 1. створити пакет, назва якого співпадає з назвою таблиці індивідуального завдання:  * в пакет включити три процедури: * CREATE\_OBJECT - додавання рядка в таблицю * MODIFY\_OBJECT - зміна значень колонок таблиці * DESTROY\_OBJECT - видалення рядка з таблиці * кожна процедура повинна мати вхідні параметри, які забезпечують керування всіма колонками таблиці  1. додати обмеження цілісності:  * унікальності назв сутностей в таблицях; * заборона на присутності цифр в строкових значеннях * заборона на наявності порожніх значень в строкових значеннях  1. додати контроль значень переданих параметрів в процедурах |
| 8 | Створення екранних форм | 1. створіть екрану форму для редагування значень однієї таблиці з індивідуального завдання, використовуючи VisualStudio та мову C#, з урахуванням 3-х операцій:додати, змінити та видалити екземпляр класу |

Продовження табл. 1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 9 | Створення класів з редагування таблиць БД | 1. створіть три програмних класи для редагування таблиці з попереднього етапу:  * додати екземпляр класу * змінити екземпляр класу * видалити екземпляр класу  1. підключіть класи до елементів екранної формипопереднього етапу |

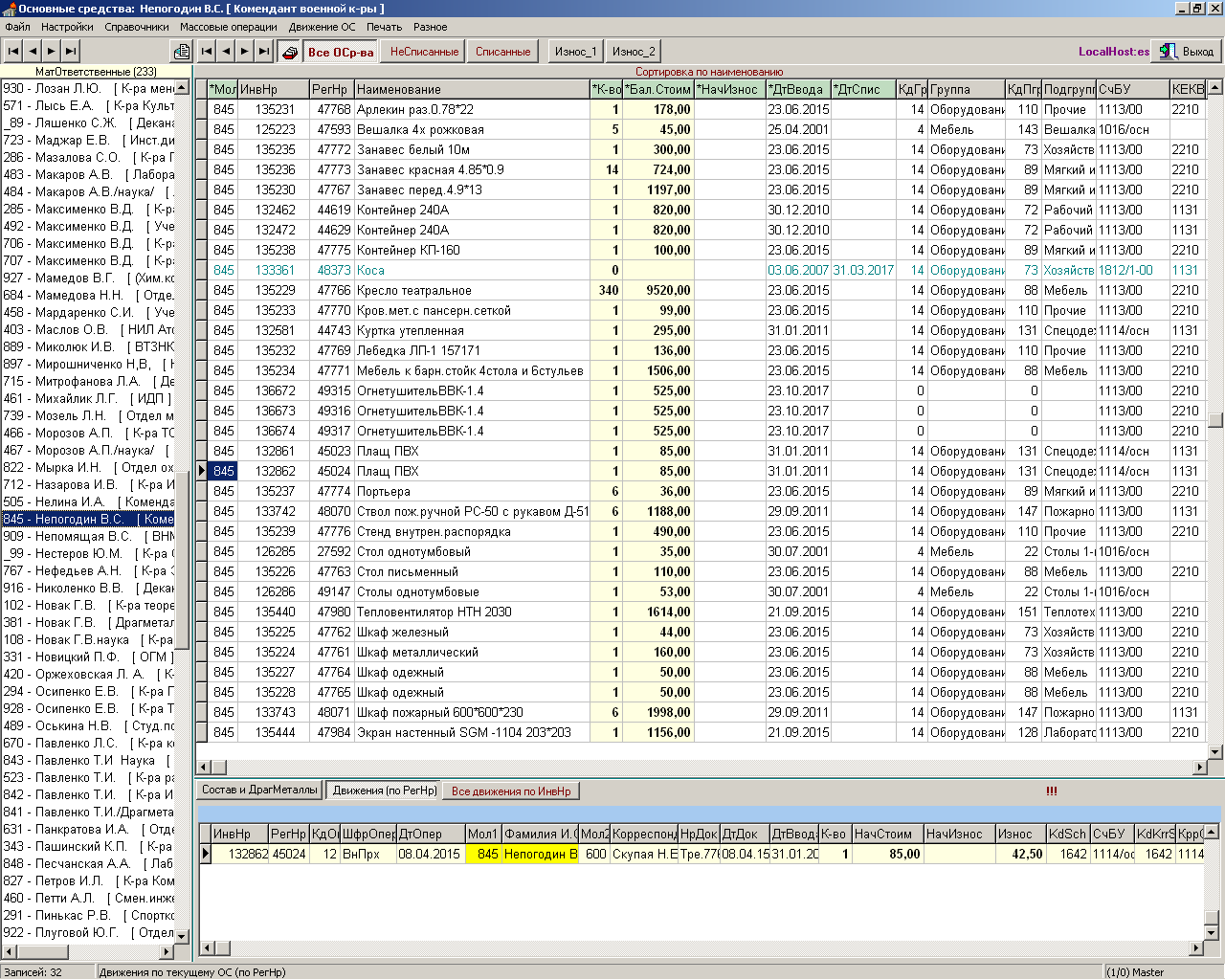
# 

**1 Моделювання предметної галузі**

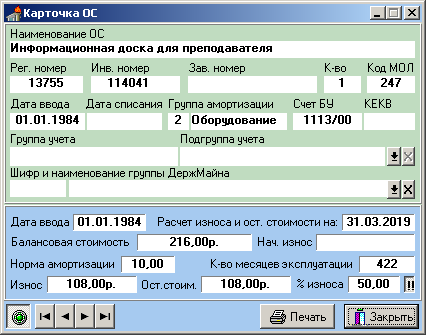
**1.1 Аналіз інтерфейсу користувача програмного модуля**

* + 1. **Призначення програмного забезпечення «Основні засоби»**

Представлені головні (основні) вікна старого програмного модуля в різних режимах функціонування(див. Рисунки 1.1-1.6)

  
Рисунок 1.1 - Головне меню (Основные средства)

Для більш детальнішої інформації - вікно КАРТКА ОС(див. Рисунок 1.2)

  
Рисунок 1.2 - КАРТКА ОС

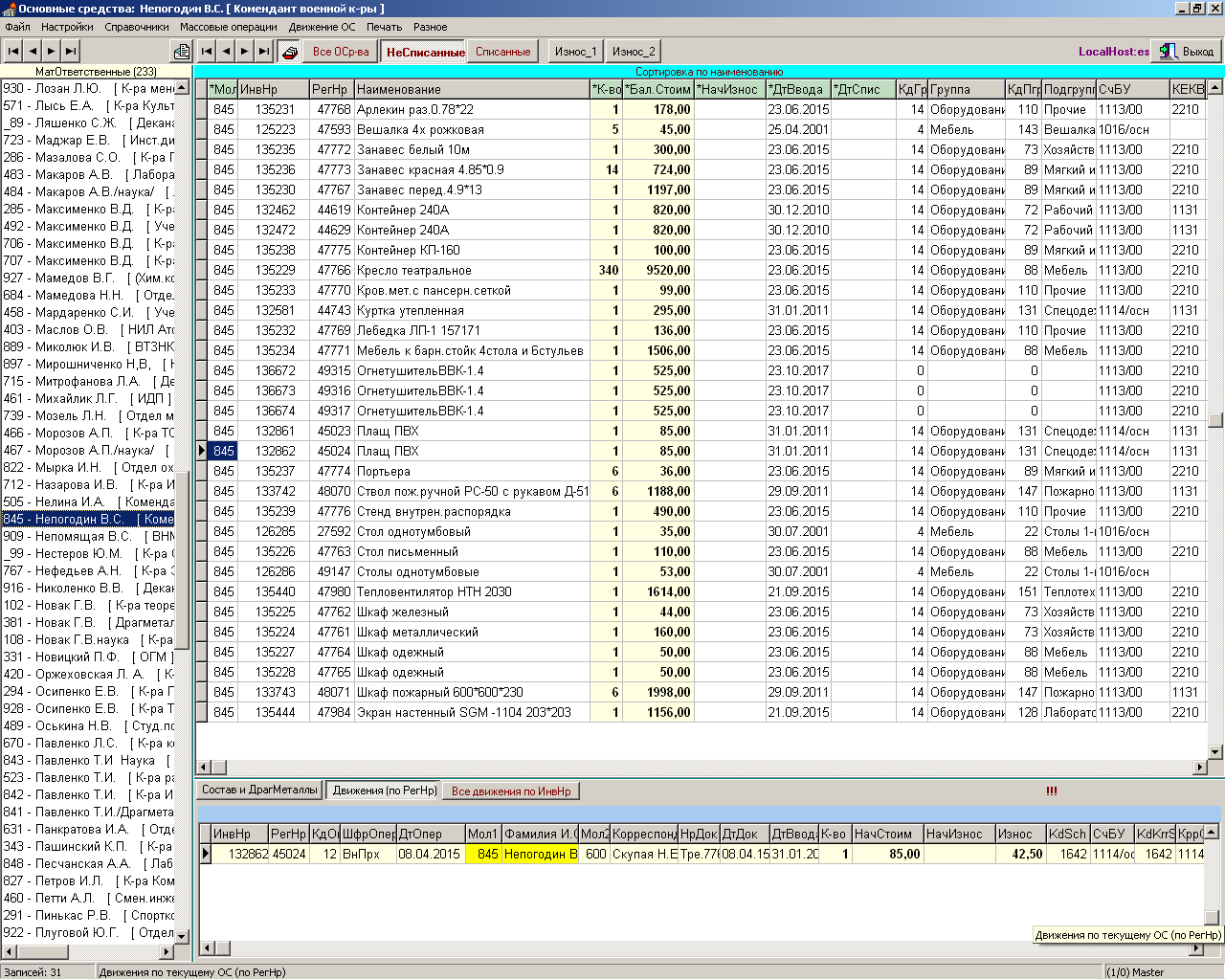
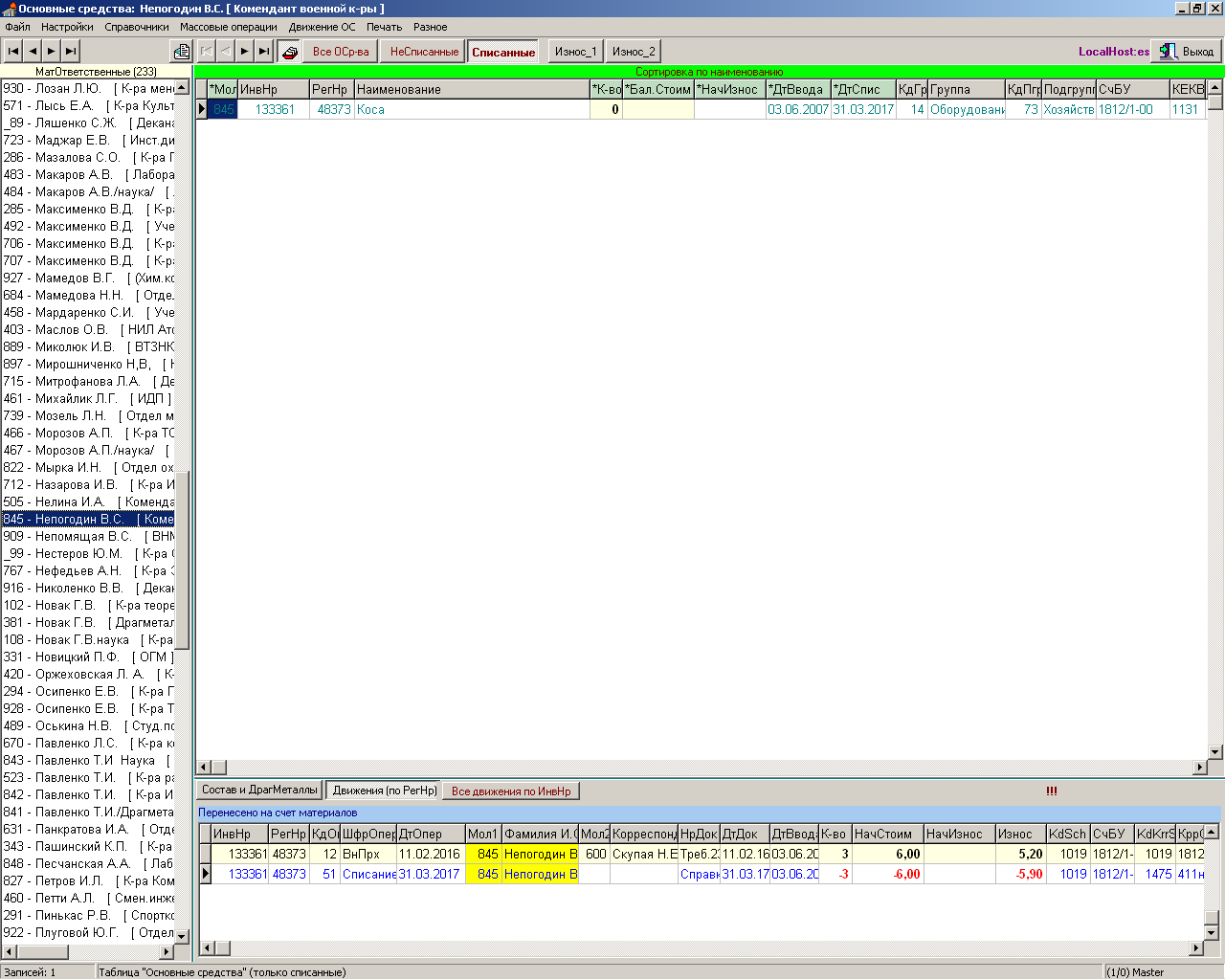
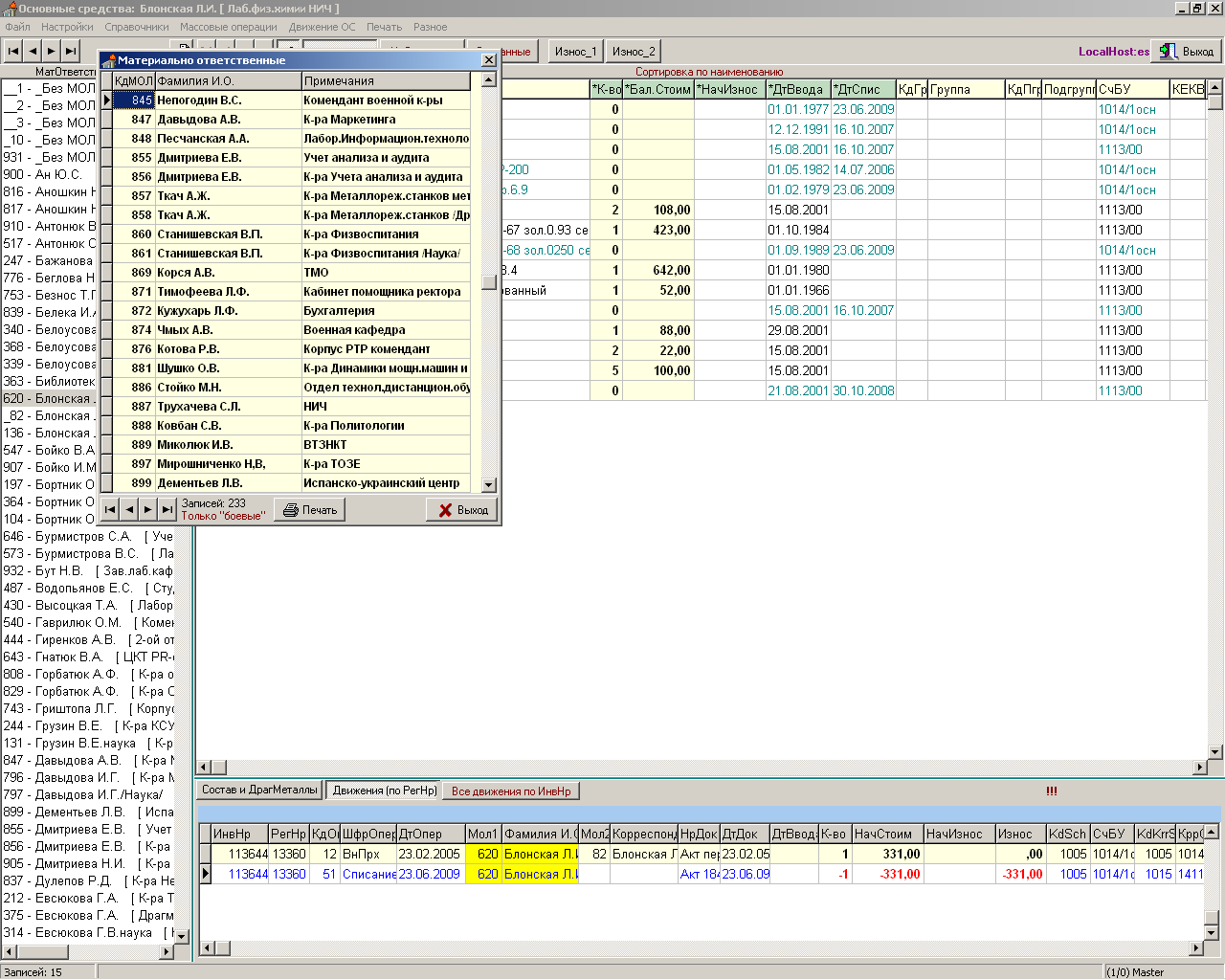
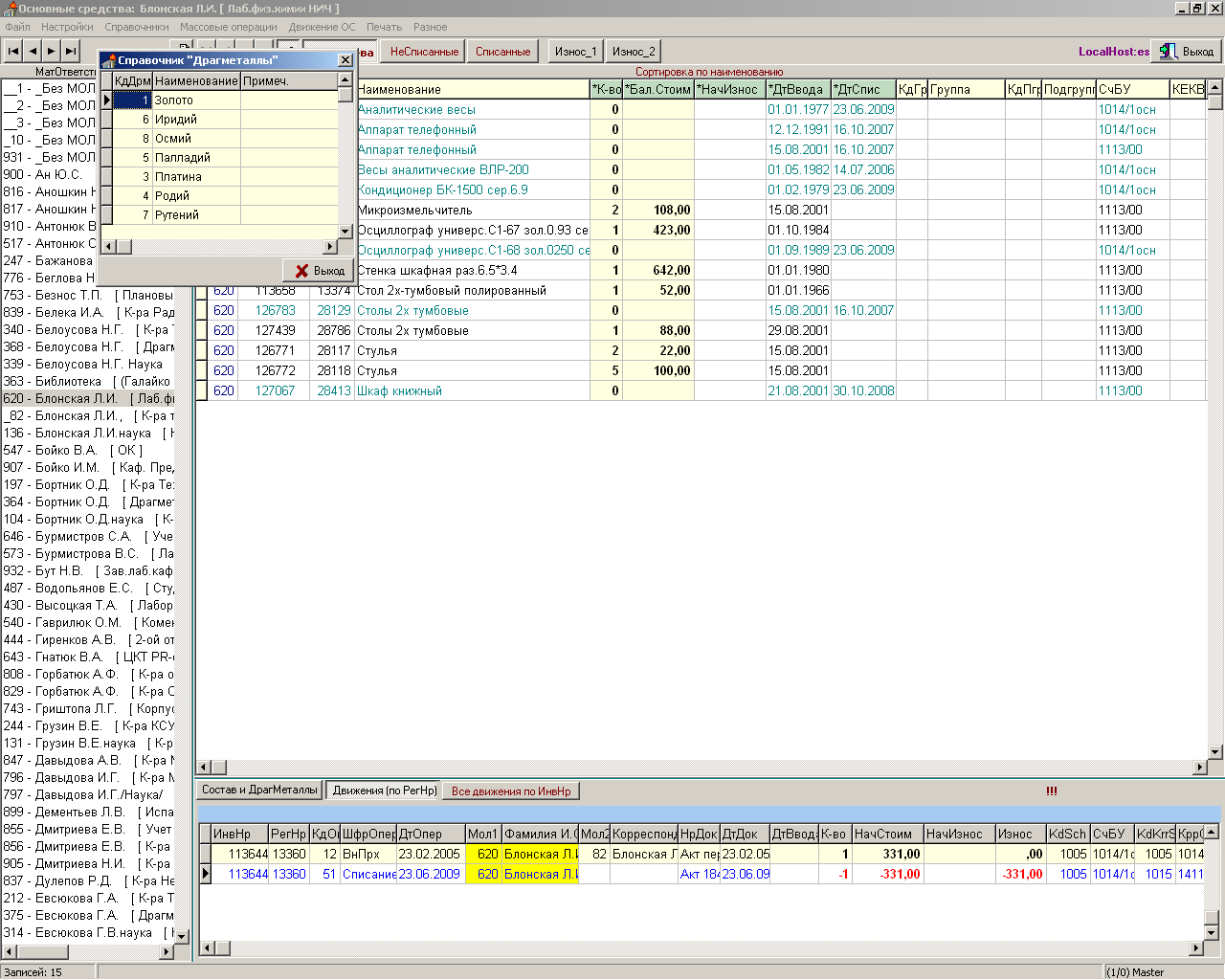


Рисунок 1.3 -Додаткове меню (Несписанные основные средтсва)

  
Рисунок 1.4 -Додаткове меню (Списанные основные средтсва)

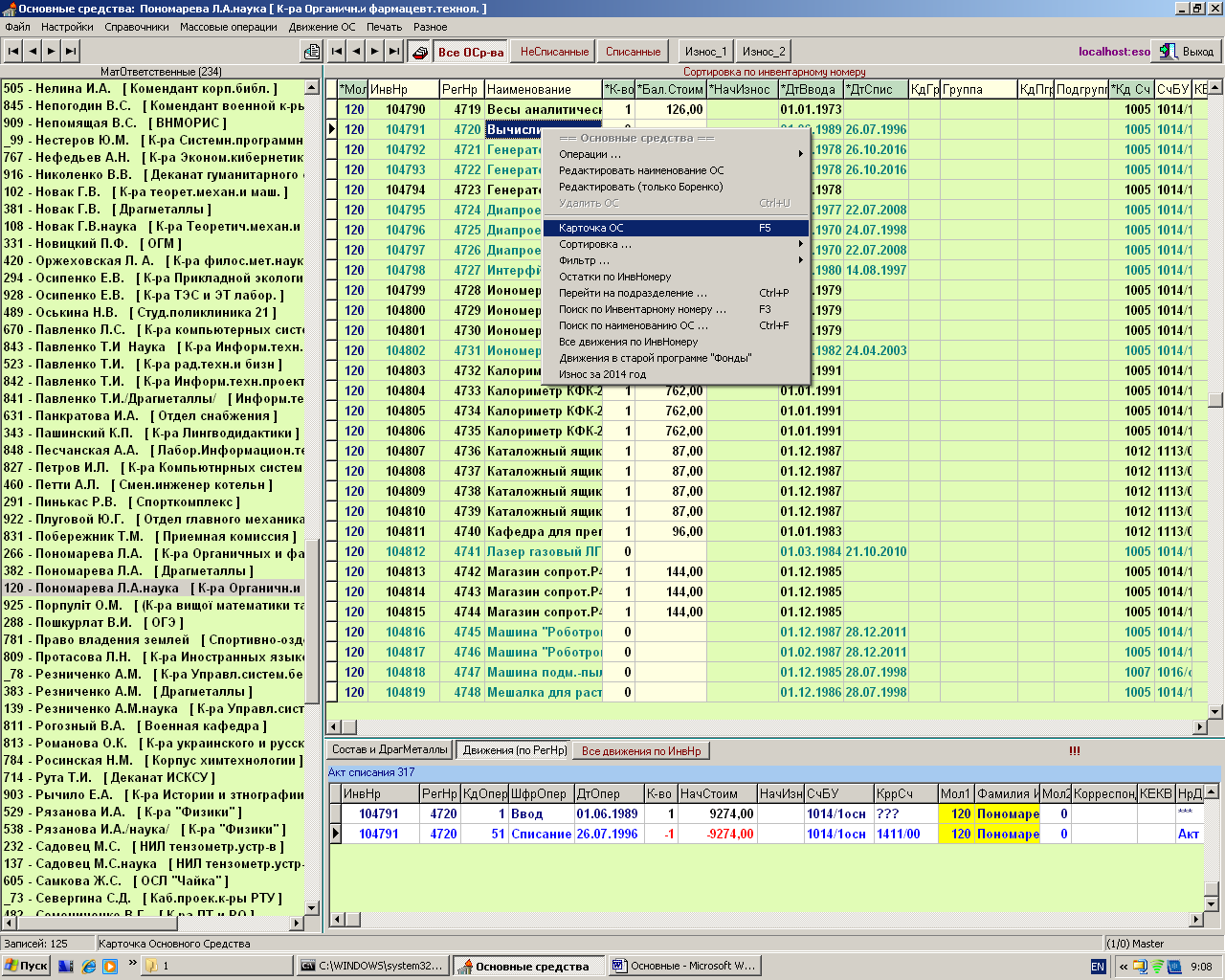
  
Рисунок 1.5 - Основне Меню (Материально ответственные - МОЛ)

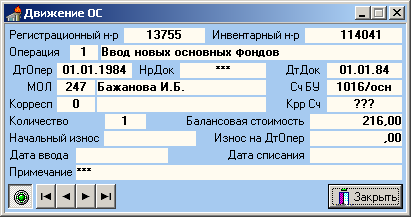
  
  
Рисунок 1.6 - Меню (Драгметаллы)

* + 1. **Коротка інструкція користувача з використання програмного забезпечення «Основні засоби»**

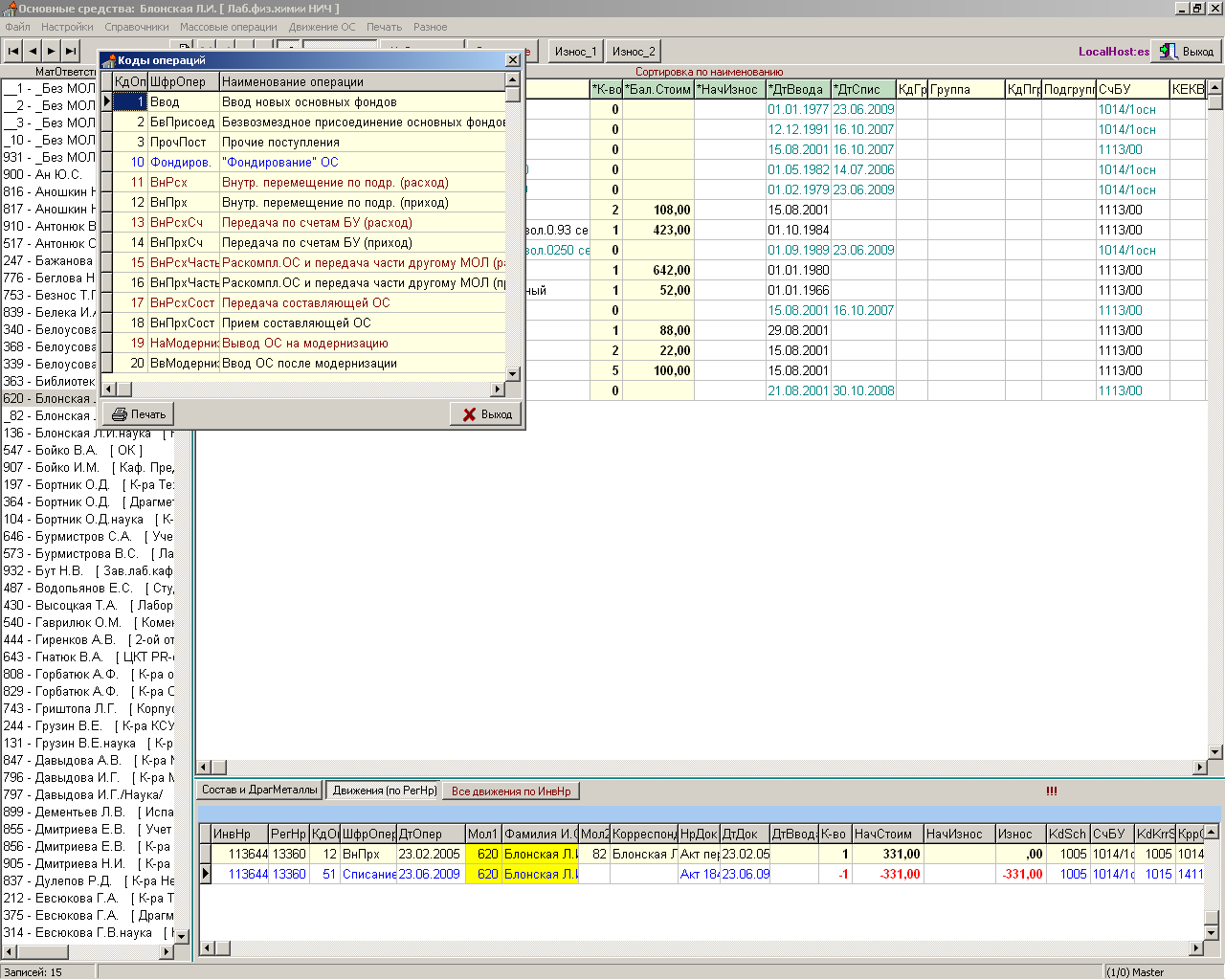
Для додавання нового запису ОС заходимо до головного меню (Основные средства) => картка ОС.

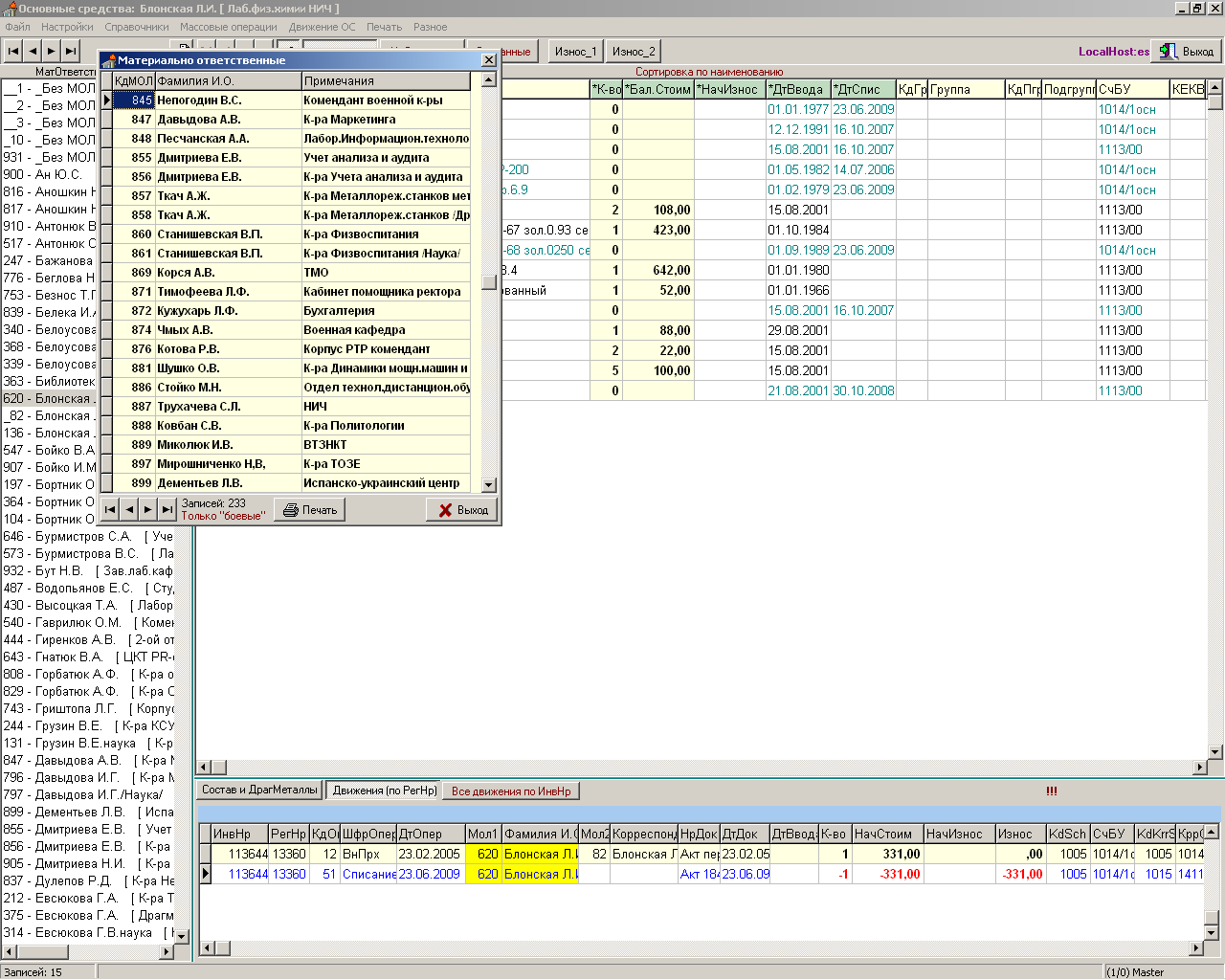
Кнопки для переключення між головними меню (Основные средства => Списанные основные средства => НеСписанные основные средства) знаходяться угорі та записані червоним кольором тексту, див. Рисунок 1.7

  
Рисунок 1.7 - Приклад екранної форми ОС  
Також можна подивитись Движение ОС(див. Рисунок 1.8)

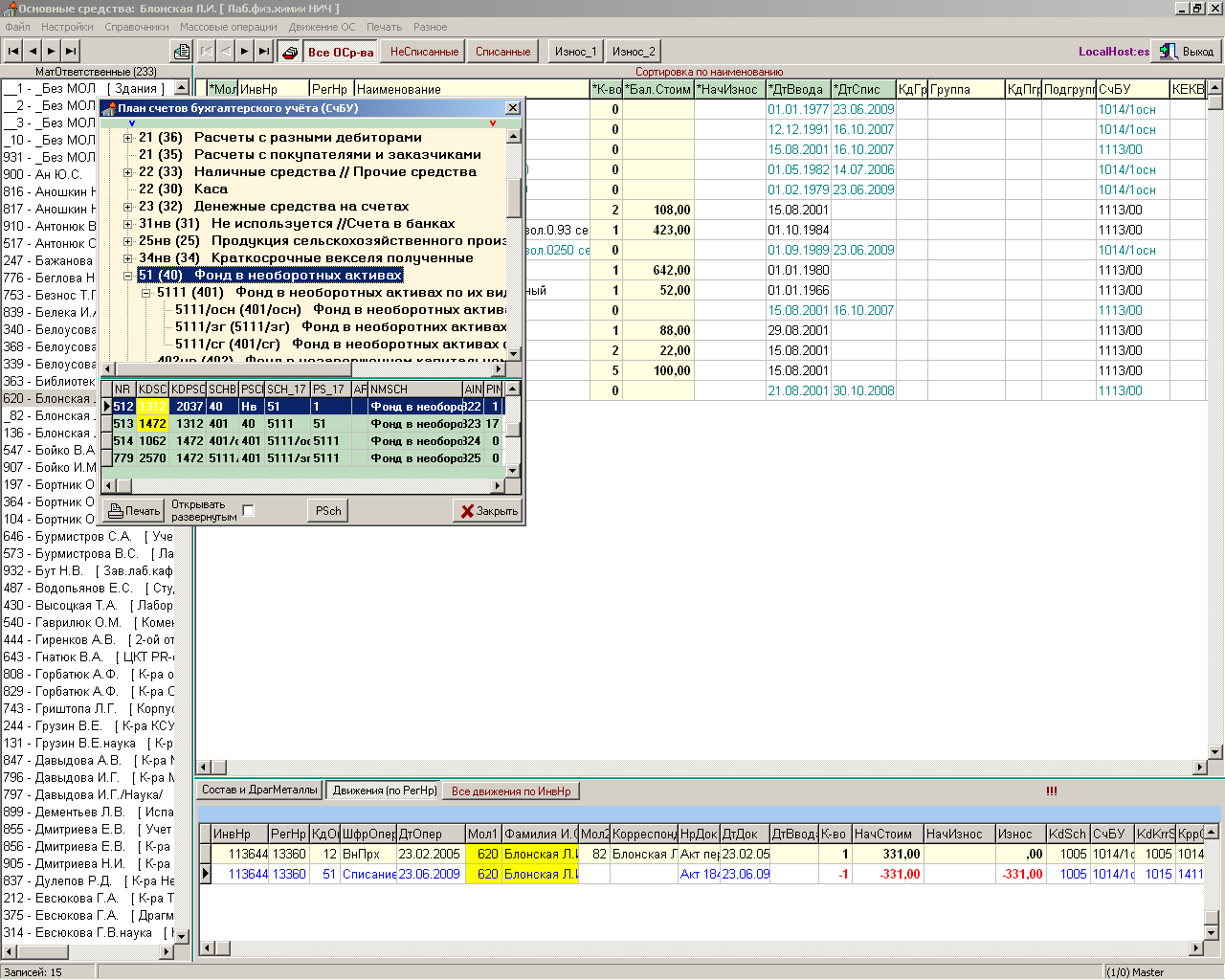
  
Рисунок 1.8– відображенння «Движения ОС»

Через меню “Справочники” можемо переходити до меню “Коды операций”, як це зображено на наступному рисунку.

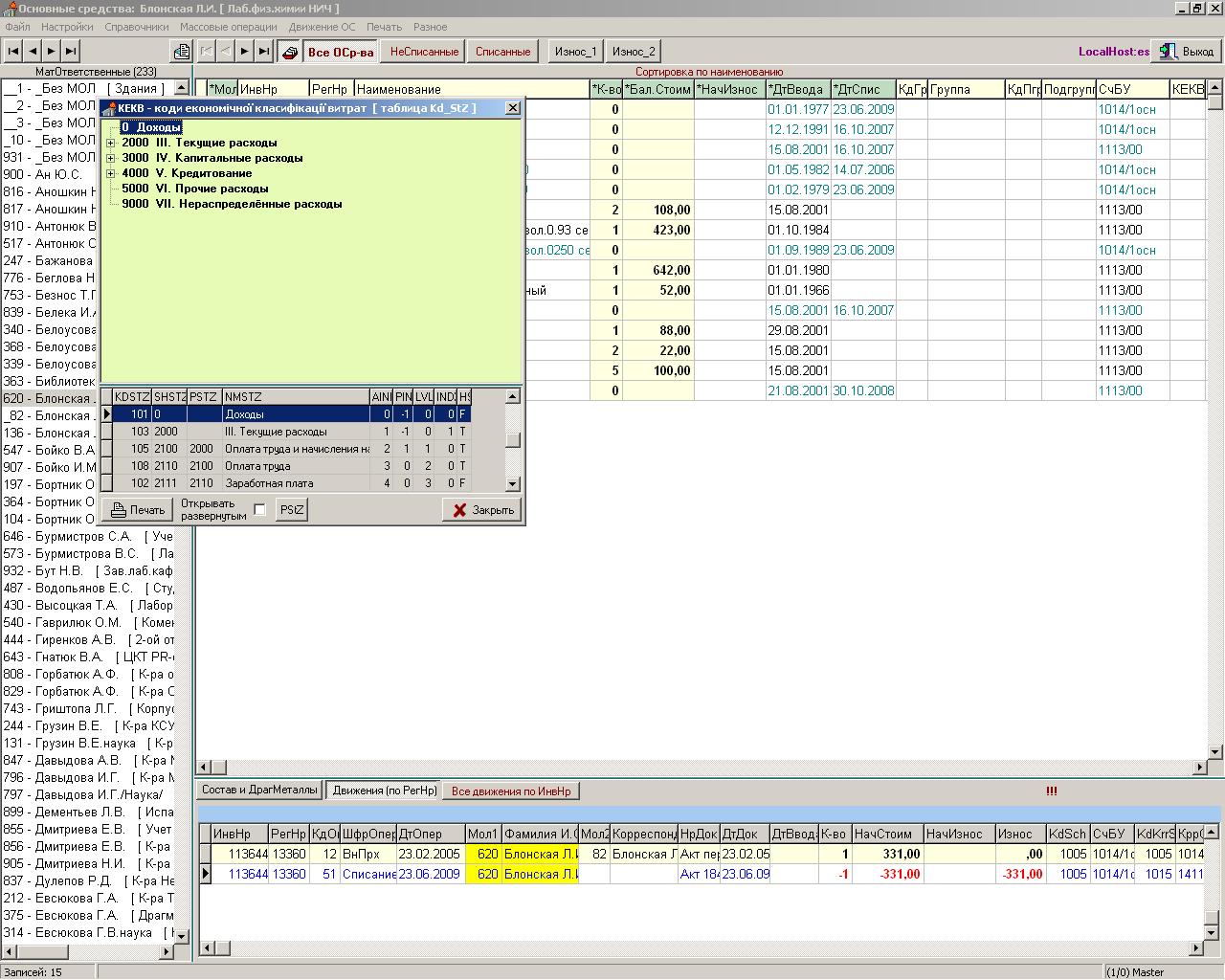
  
Рисунок 1.9– відображенння «Коди операцій»

За допомогою меню “Справочники” можемо переходити до меню “МОЛ – материально ответственные”(див. Рисунок 1.10)  
Рисунок 1.10 – відображенння меню “МОЛ – материально ответственные”

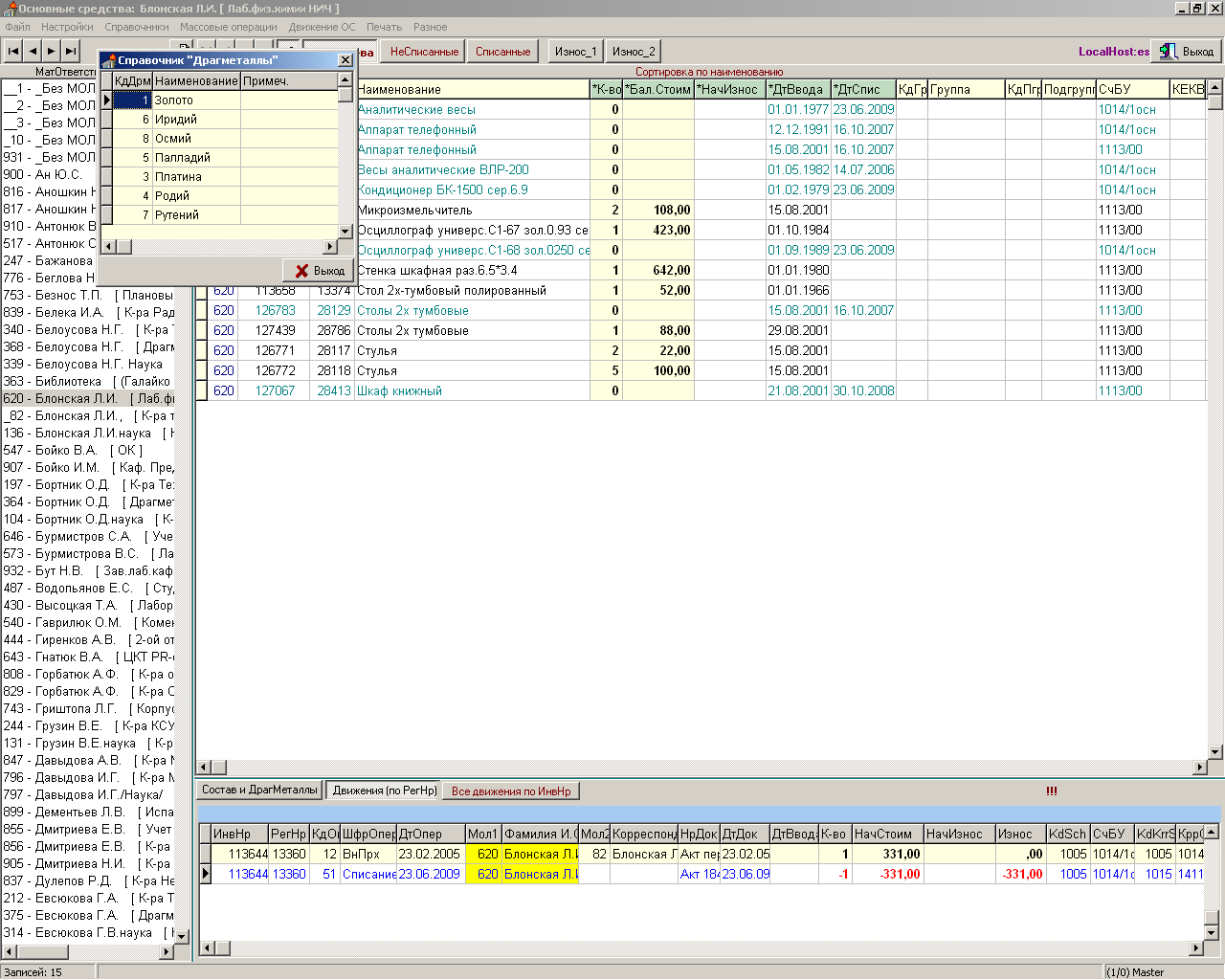
За допомогою меню “Справочники” можемо переходити до меню “План счетов бухгалтерского учета”

  
Рисунок 1.11 - меню “План счетов бухгалтерского учета”

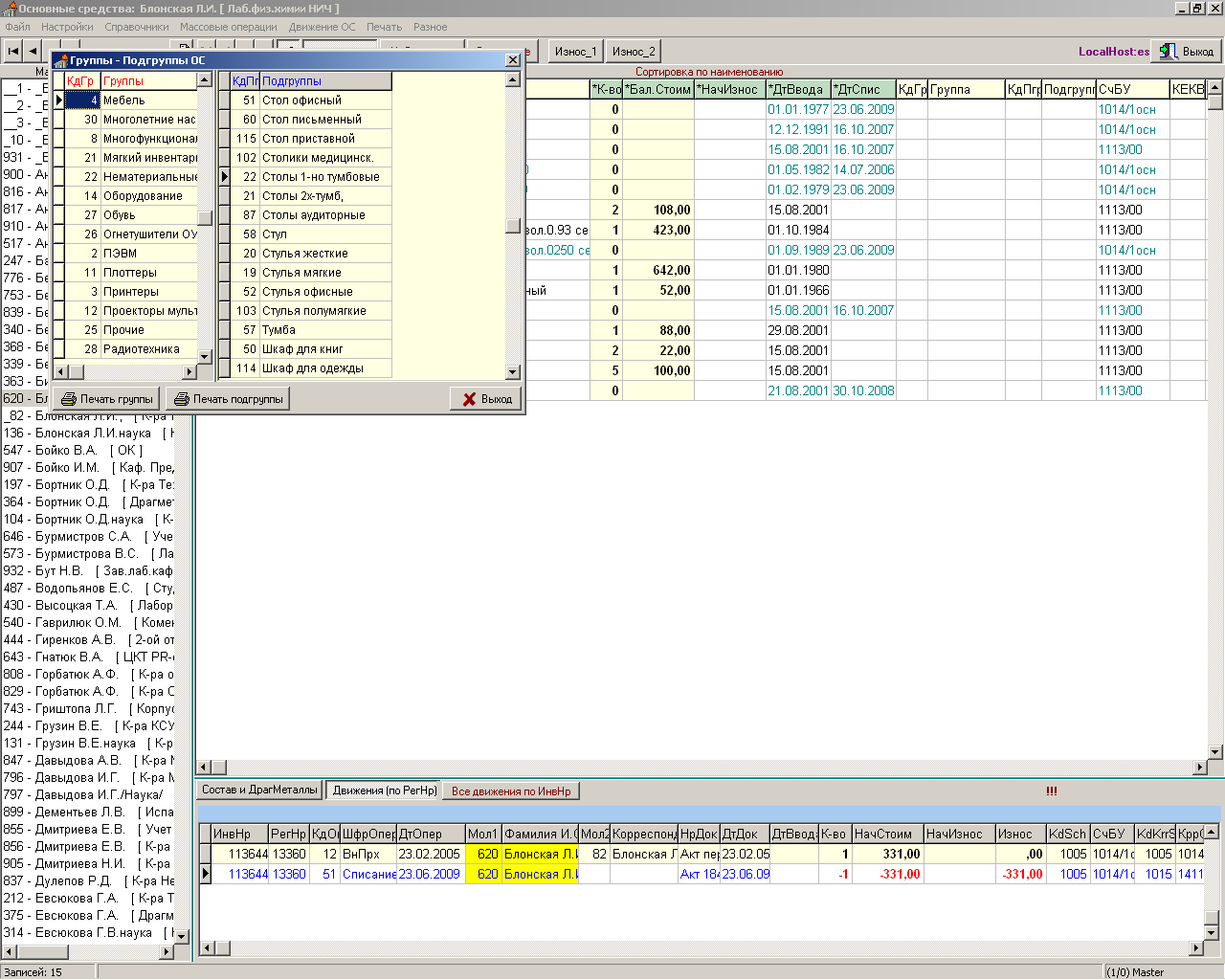
Меню “Справочники” дає змогу переходити до меню “Виды затрат (KEKB)”

  
Рисунок 1.12 – відображенння меню “Виды затрат (KEKB)”

Також можемо переходити до меню “Драгметаллы”(див. Рисунок 1.13)

  
Рисунок 1.13 – відображенння меню “Драгметаллы”

Теж меню “Справочники” забезпечую перехід до меню “Группы / Подгруппы ”

  
Рисунок 1.14 – відображенння меню “Группы / Подгруппы ”

Використовуючи меню “Справочники”, можемо переходити до меню “Клиенты”

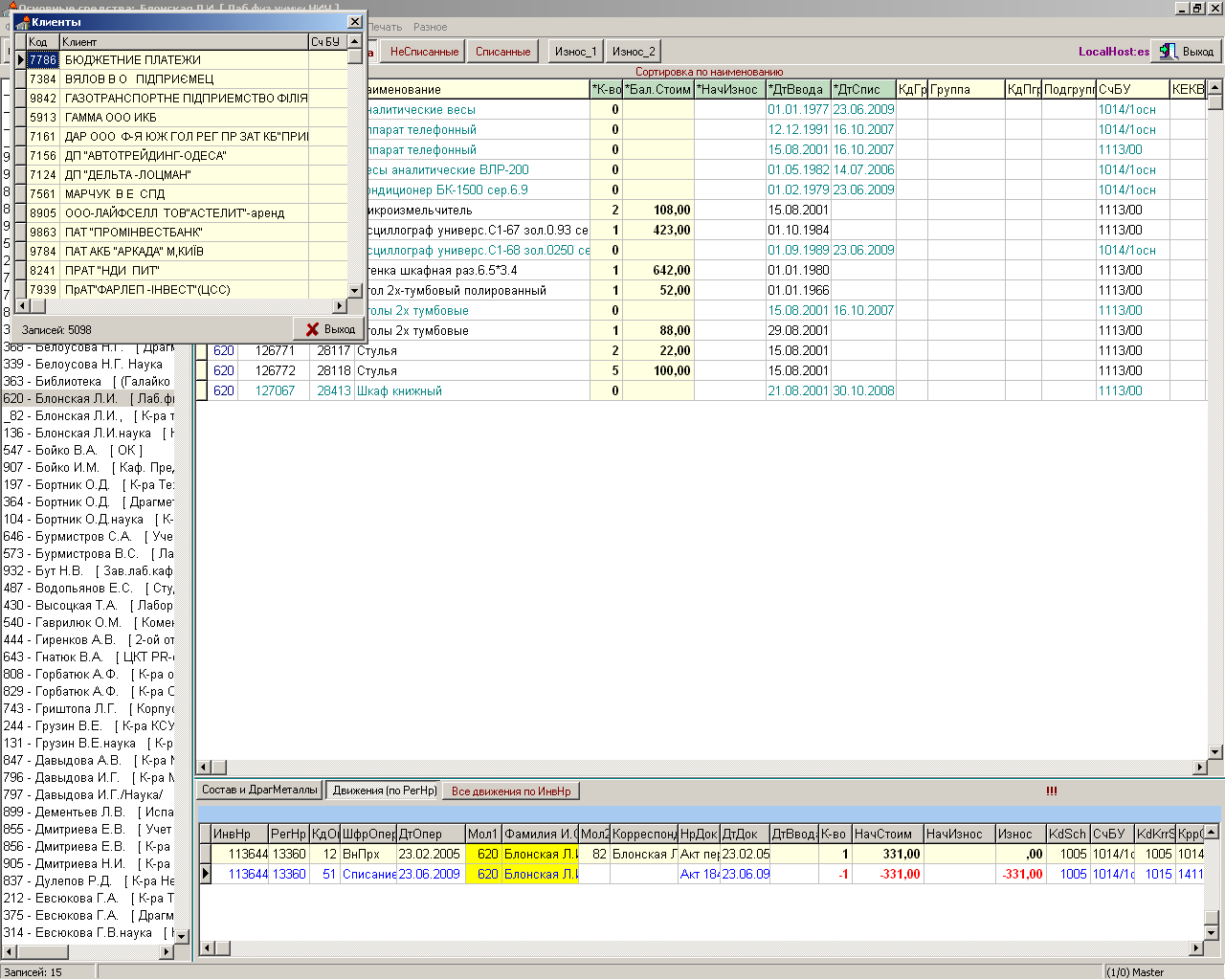


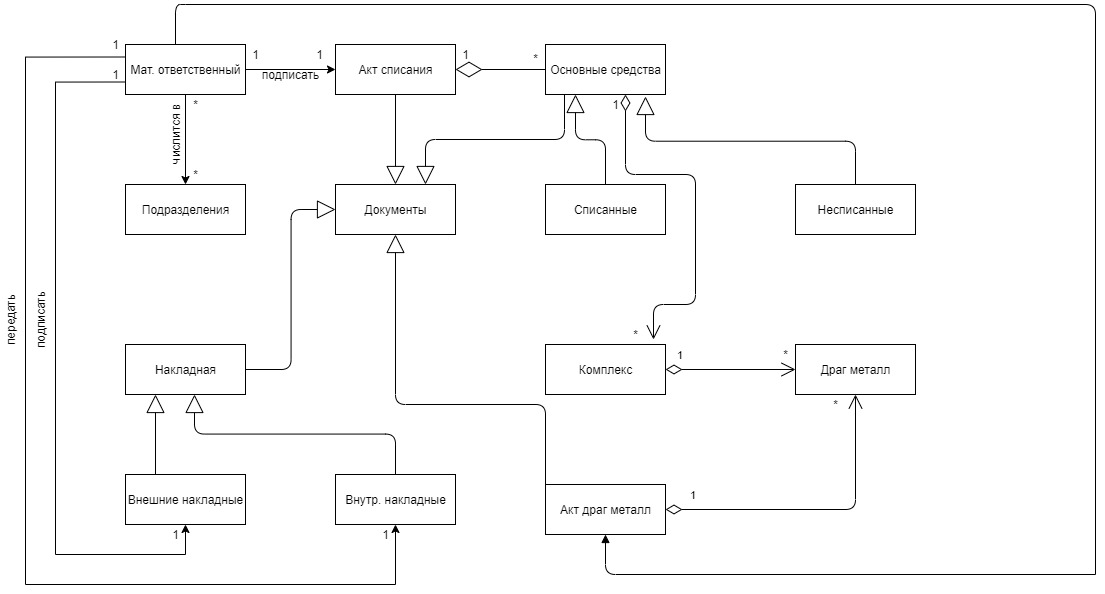
Рисунок 1.15 – відображенння меню “Клиенты”

* 1. **Моделювання структур даних**
     1. **Опис назв об`єктів та їх атрибутів на основі аналізу інструкцій користувача по роботі з програмним забезпеченням**

**Мат. Ответственный: id,** *фамилия, имя, отчество* **Акт списания: id,** *дата, название* **Основные средства: id,** *название* **Подразделения:id,** *название* **Документы:id,** *название* **Списанные осн. Средства: id,** *название* **Несписанные осн. Средства: id,** *название* **Накладная: id,** *дата, название* **Комплекс:id,** *название* **Драг металл:id,** *название* **Внешняя накладная: id,** *дата, название* **Внутр. Накладная: id,** *дата, название* **Акт драг. металл: id,** *дата, название*

* + 1. **Опис UML-діаграми концептуальних класів**

Визначивши об’єкти та атрибути, створюємо UML-діаграму концептуальних класів(див. Рисунок 1.2.2)

  
Рисунок 1.2.2 – UML-діаграма концептуальних класів

Згідно з індивідуальним завданням, визначимо назви класів для роботи з ними:

**- Документи  
- Подразделение  
- Внутренняя накладная**

**2 Розробка бази даних**

**2.1 Проектування реляційної моделі бази**

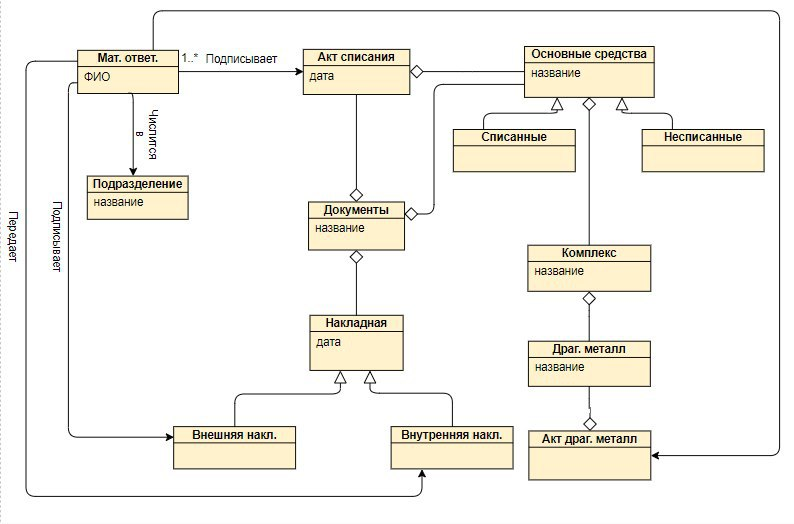
Створюємо реляційну модель даних на основі UML-діаграми концептуальних класів

Рисунок 2.1 – Приклад реляційної моделі БД

**2.1.1 Опис реляційної моделі даних**

|  |
| --- |
| **Внутренняя накладная** |
| id |
| Name |
| Data |

|  |
| --- |
| **Документы** |
| id |
| Name |
| Data |

|  |
| --- |
| **Подразделения** |
| id |
| Name |

**2.1.2. Опис типів даних атрибутів таблиць реляційної моделі**

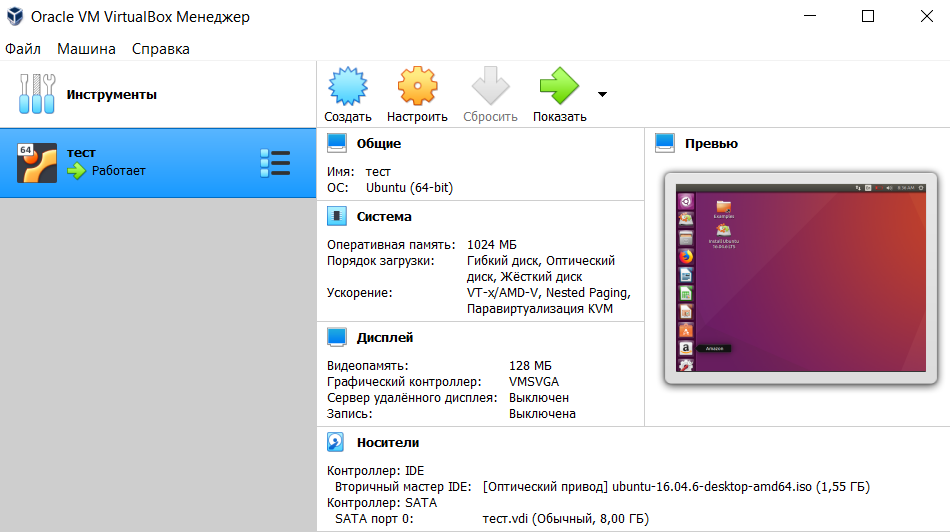
**Внутренняя накладная:** *id ( integer AUTO INCREMENT), name VARCHAR(20), data (DATE)*

**Документы:** *id ( integer AUTO INCREMENT), name VARCHAR(20), data (DATE)*

**Подразделения:** *id ( integer AUTO INCREMENT), name VARCHAR(20)*

## **2.2. Опис процесу інсталяція СКБД FireBird в ОС Linux**

### **2.2.1. Встановлення ОС Linux**

  
Рис 2.2.1 – приклад встановлення ОС Linux

Створюємо користувача з правами адміністратора(назва співпадає з прізвищем у транслітерації)

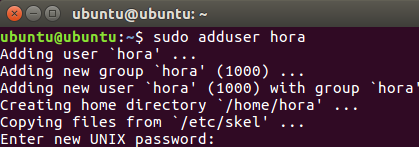
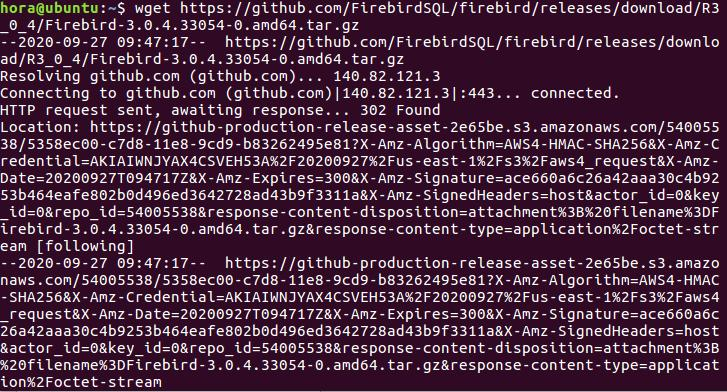


Рис. 2.2.2. – Приклад створення користувача з правами адміністратора

### **2.2.2. Описати кроки інсталяції СКБД FireBird з прикладами кроків у вигляді скріншотів екранів**

Отримуємо установочний пакет дистрибутива Firebird 3.0.4(рис. 2.2.2.1-2.2.2.4)

 Рис. 2.2.2.1 – Приклад інсталяції СКБД FireBird

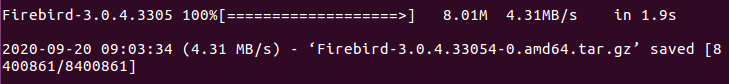


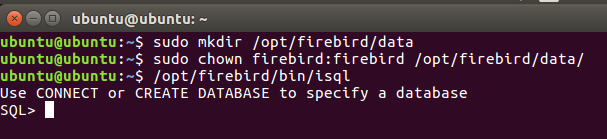
Рис. 2.2.2.2 - Разархівація пакету

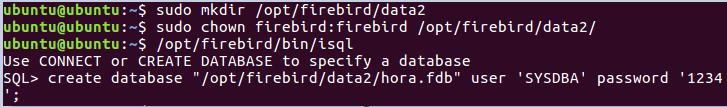
 Рис. 2.2.2.3 – Завантаження додаткового пакету



Рис. 2.2.2.4 – Завантаження основного пакету

**2.2.3.** **Опис кроків створення БД в СКБД FireBird**

  
Рис 2.2.3.1. – Приклад використовування утиліти isql

  
Рис 2.2.3.2 – Створення директорії для БД

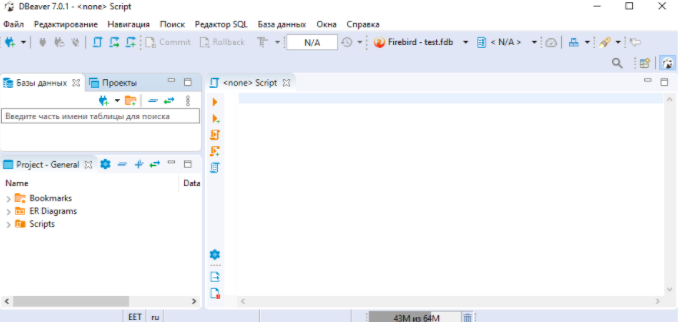
**2.2.4. Опис кроків налаштування доступу до БД з використанням DBeaver**

Рис 2.2.4.1 – приклад основної екранної форми програми

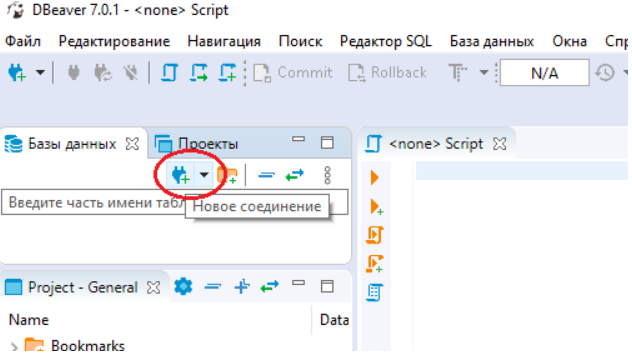


Рис 2.2.4.2. – Приклад екранної форми початку процесу налаштування з’єднання

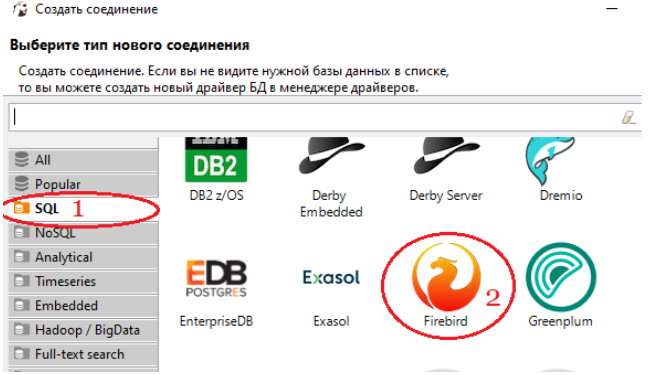


Рис 2.2.4.3. – Приклад екранної форми з вибором типу нового з’єднання

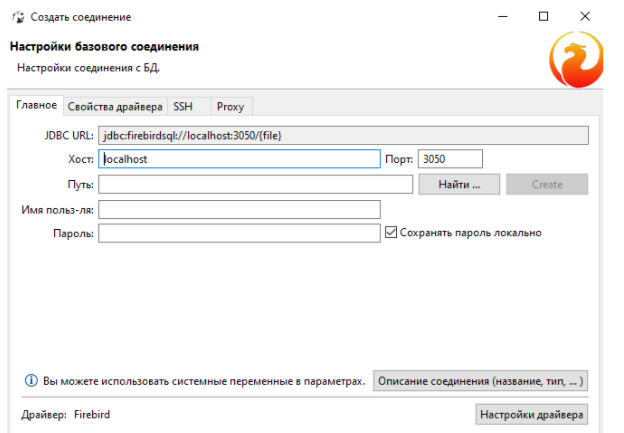


Рис 2.2.4.4. – Приклад екранної форми з полями налаштування нового з’єднання

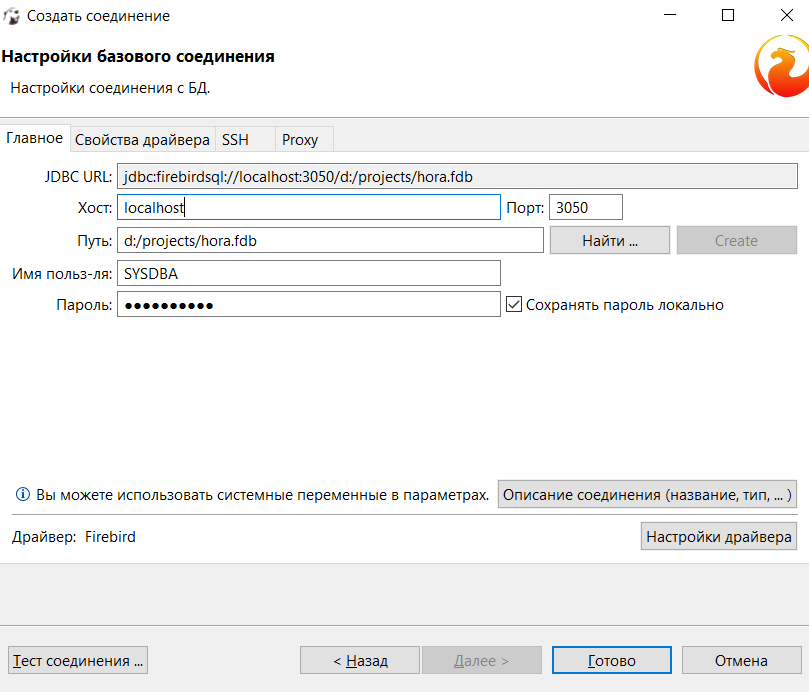


Рис 2.2.4.5. – приклад екранної форми зі внесеними значеннями в поля налаштування нового з’єднання

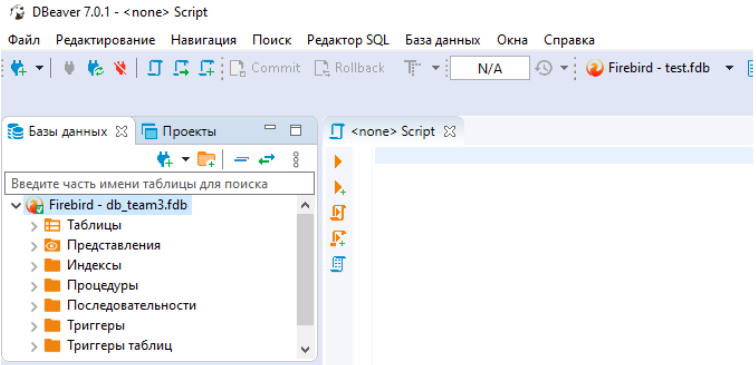


Рис 2.2.4.6. – Початок роботи з СКБД

## **Конструювання БД в СКБД FireBird**

**2.3.1. Опис SQL-скрипта створення структури БД**

Створюємо таблиці «Документи», «Внутрішня накладна», «Підрозіл»

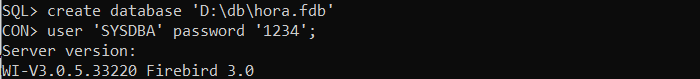
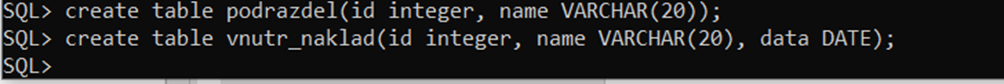
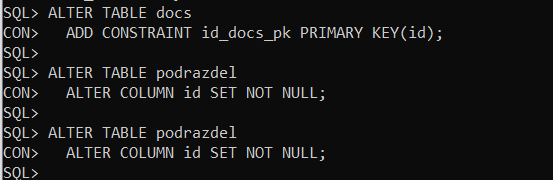


Рис. 2.3.1.1. – Приклад створення таблиць

Заборона нульвого значення й установка первинного ключа в поле id таблиць

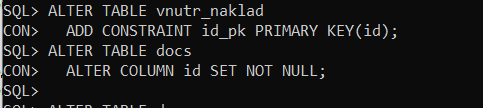
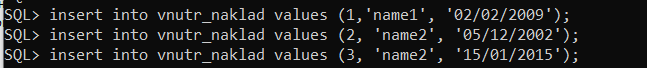


Рис. 2.3.1.2 – Приклад заборони нульвого значення й установка первинного ключа в поле id таблиці «Накладная»

**2.3.2 Опис SQL-скрипта заповнення таблиць БД тестовими даними**

Створити скрипт із прикладами SQL-команд внесення даних дотаблиць БД з індивідуального завдання(10 INSERT-команд)

Заповнення таблиць:

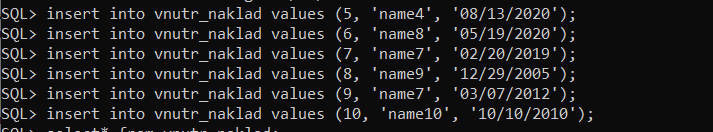


Рис. 2.3.2.1 – Приклад insert команд для таблиці «Внутрішня накладна»

Insert команди для таблиць «Документи» та «Підрозділ»

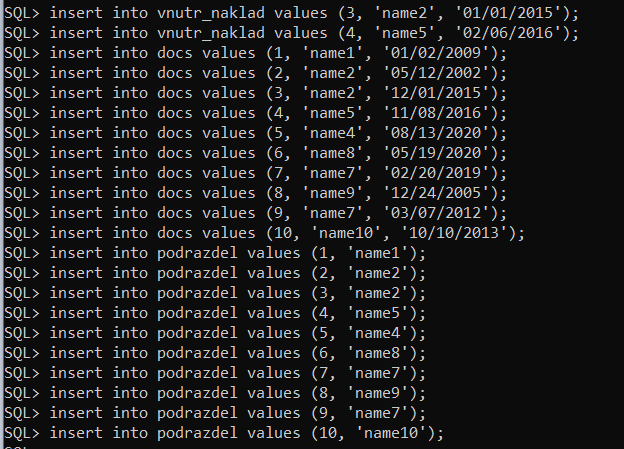


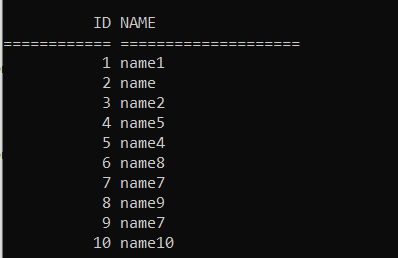
Рис. 2.3.2.2. – Приклад Insert команд для таблиць «Документи» та «Підрозділ»  
Створимо скрипт із прикладами SQL-команд зміни даних

в таблицях БД

Редагування даних в таблиці «Подраздел»



Рис. 2.3.2.3 – приклад Update команди для таблиці «Подраздел»

  
Рис. 2.3.2.4 – Результат Update команди для таблиці «Подраздел»

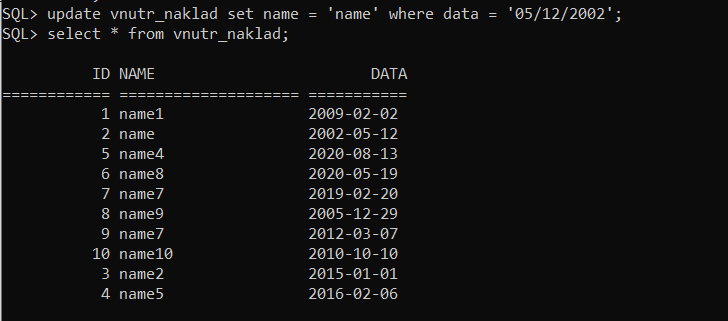
  
Рис. 2.3.2.5 – Редагування даних в таблиці «Внутрішня накладна»

Рис. 2.3.2.6 – Редагування даних в таблиці «Документи»

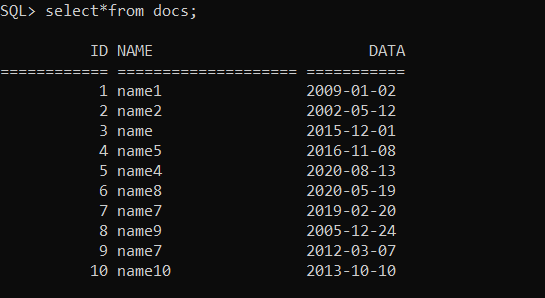


Рис. 2.3.2.7 – Результат Update команди для таблиці «Документи»

Створимо скрипт із прикладами SQL-команд видалення

даних з таблиць БД:

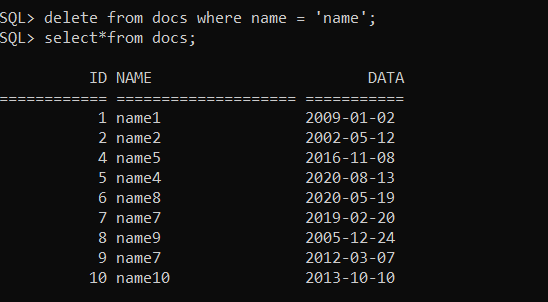
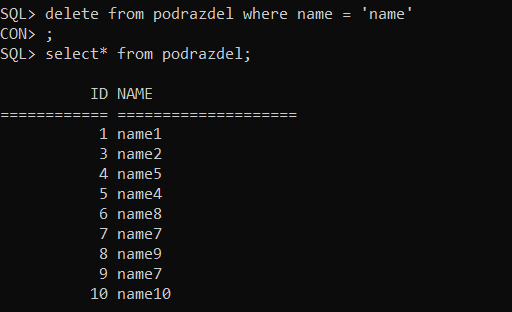


Рис. 2.3.2.8 – Результат Delete команди для таблиці «Документи»

  
Рис. 2.3.2.9 – Результат Delete команди для таблиці «Подраздел»

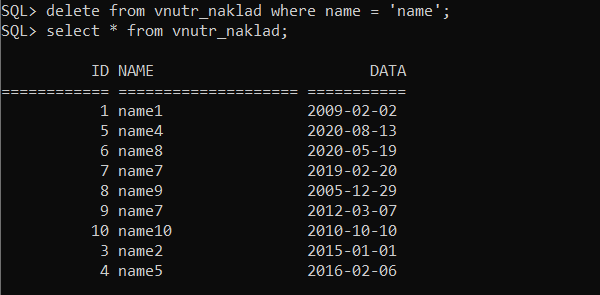


Рис. 2.3.2.10 – приклад Delete команди для таблиці «Внутренняя накладная»

|  |  |
| --- | --- |
| **2.4 Опис пакетів збережених процедур FireBird** |  |

Cтворимо пакет, назва якого співпадає з назвою таблиці індивідуального завдання:

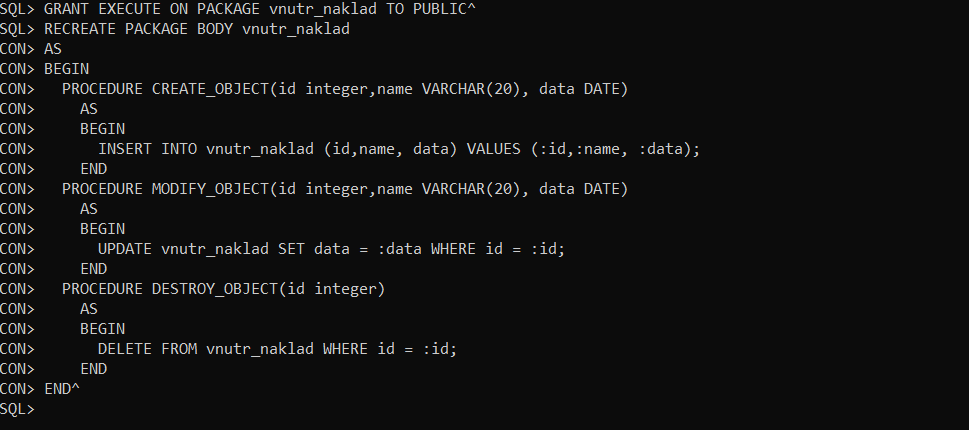
В пакет включимо три процедури:

−CREATE\_OBJECT -додавання рядка в таблицю;

−MODIFY\_OBJECT -зміна значень колонок таблиці;

−DESTROY\_OBJECT -видалення рядка з таблиці − кожна процедура має вхідні параметри, які забезпечують керування всіма колонками таблиці;

Приклад створення пакету з процедурами та їх налаштування див. на рис 2.4.1

  
Рис.2.4.1– Створення пакету з процедурами та їх налаштування

Виклик та перевірка процедур:

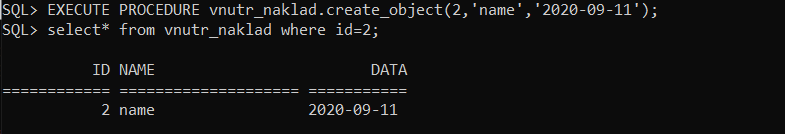


Рис. 2.4.2 – Приклад перевірки роботи процедур

Надалі додамо обмеження цілісності:

− унікальності назв сутностей в таблицях;

−заборона на присутності цифр в строкових значеннях;

− заборона на наявності порожніх значень в строкових значеннях;

Встановимо обмеження ввода цифр в таблиці «Внутрення накладная»( поле «name»)

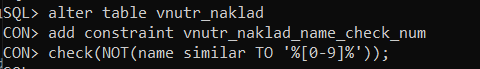


Рис. 2.4.3 – Приклад встановлення обмежень ввода цифр в таблиці «Внутрення накладная»на поле «name»

Установимо обмеження ввода пустого рядку цифр в таблиці «Внутрення накладная»( поле «name»)

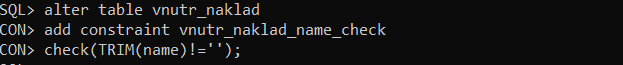
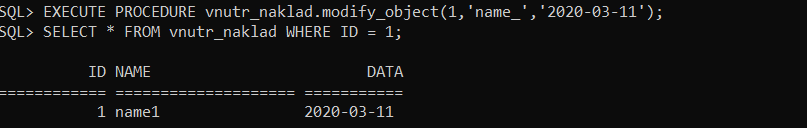
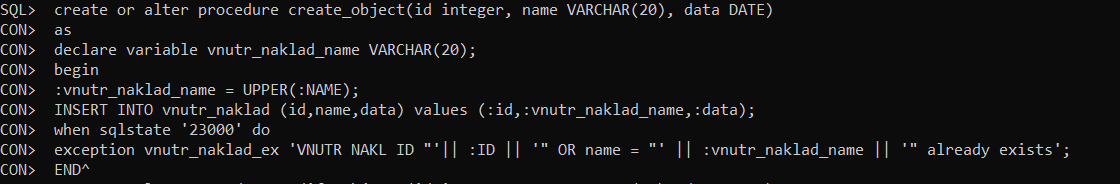
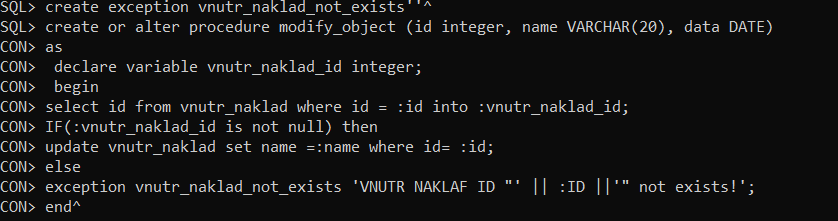


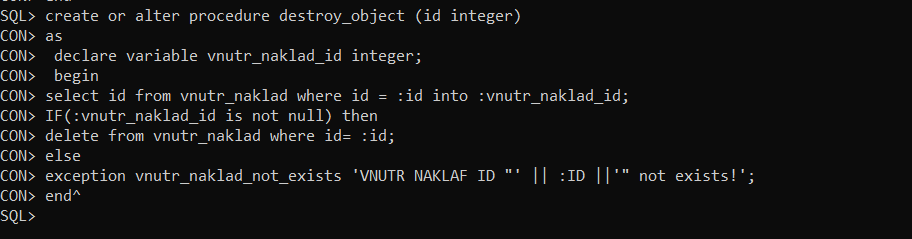
Рис. 2.4.4 – Приклад заборони на наявності порожніх значень в строкових

Значеннях в таблиці «Внутрення накладная»( поле «name»)

Тепер додамо контроль значень переданих параметрів в процедурах(див. рис. 2.4.5.-2.4.7)

  
Рис. 2.4.5 – Перевірка на існування унікальних полей

  
Рис. 2.4.6 – Перевірка на існування запису

  
Рис. 2.4.7 – Перевірка на існування запису

# **3 Розробка графічного інтерфейсу доступу до БД**

**3.1. Опис екранних форм в програмному середовищі VisualStudio**

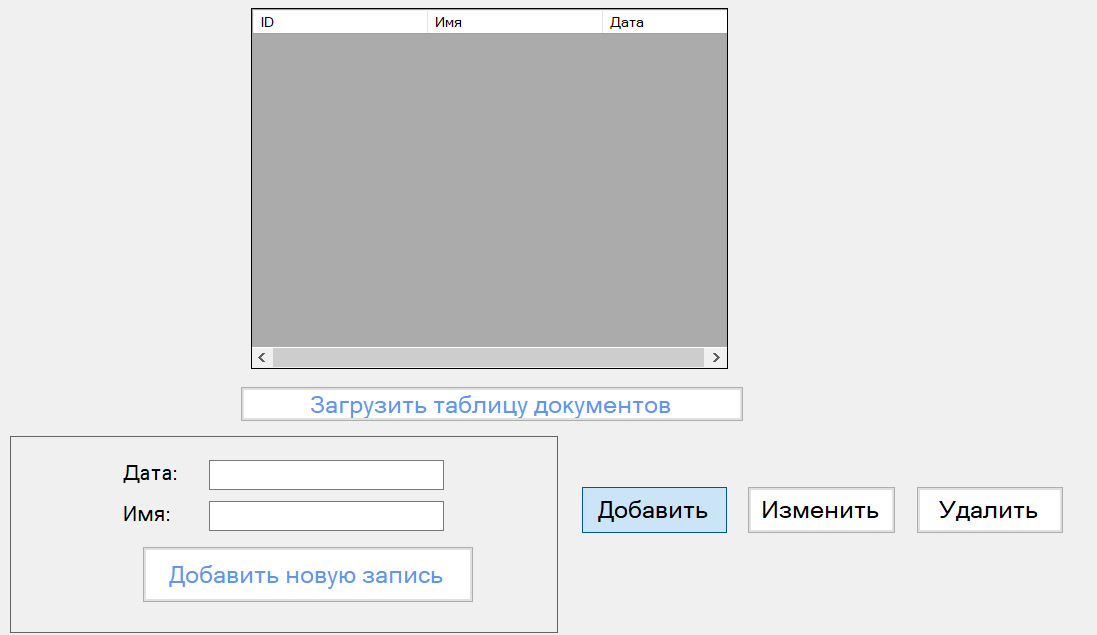


Рис. 3.1.1 – Приклад основної екранної форми

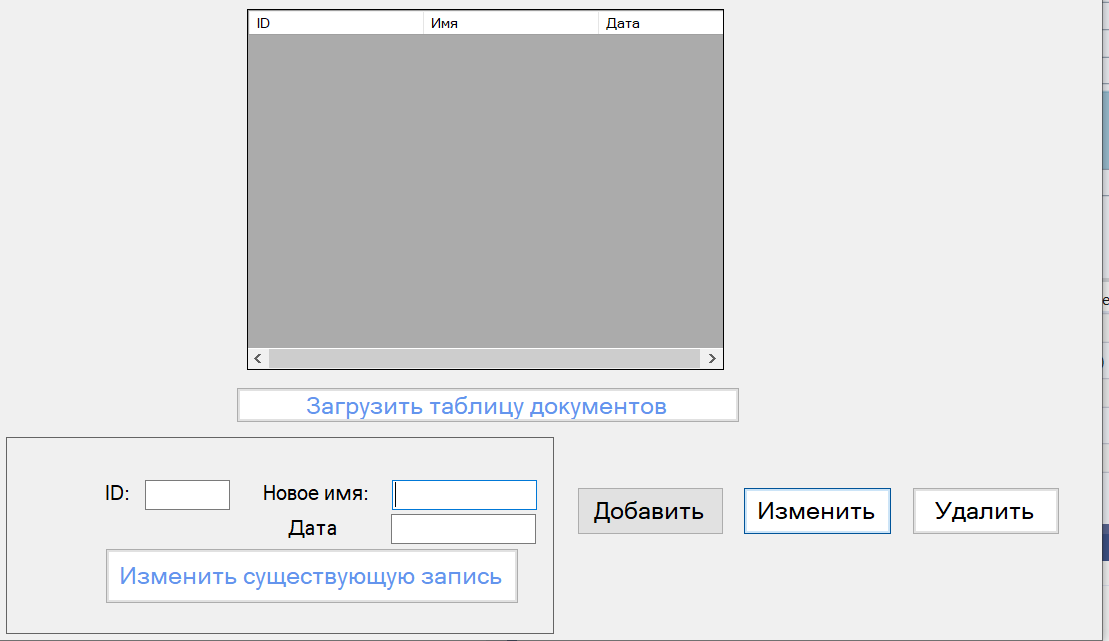
Програмний модуль в режимі редагування інформації з бази даних: 

Рис. 3.1.2 – Приклад програмного модуля в режимі редагування інформації за бази даних

Програмний модуль в режимі видалення інформації з бази даних:

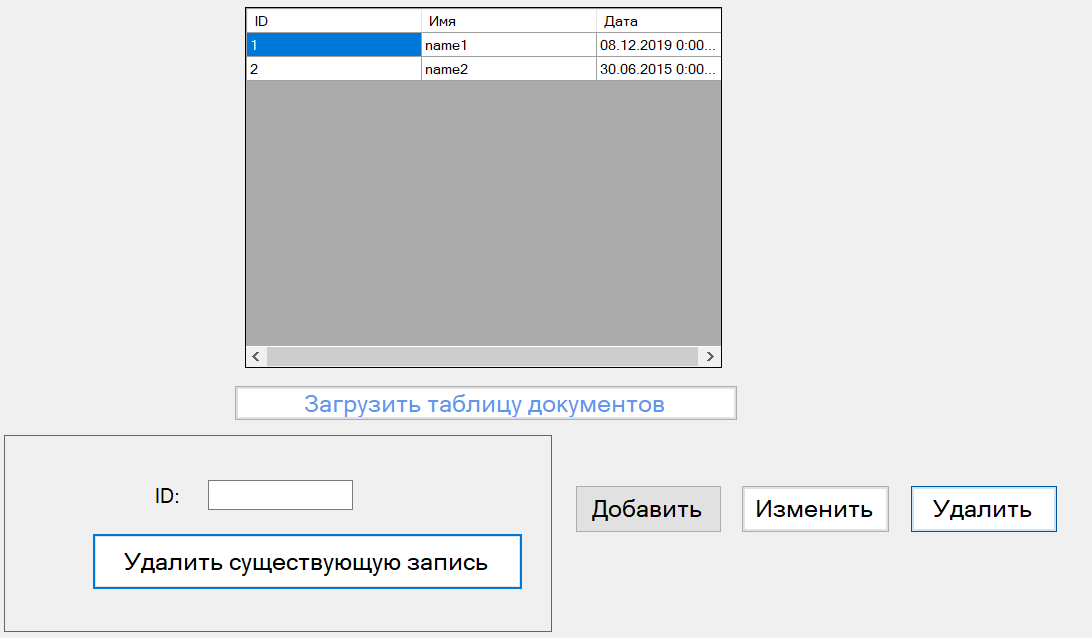


Рис. 3.1.3 – Приклад екранної форми в режимі видалення інформації з бази даних

Додавання інформації до бази даних:

Заповнюємо поля вводу даних(Датя, Имя), які відповідають за інформацію нового запису в таблиці, після чого натиснути «додати новий запис» - основна таблиця буда оновлена згідно зі введеними даними.

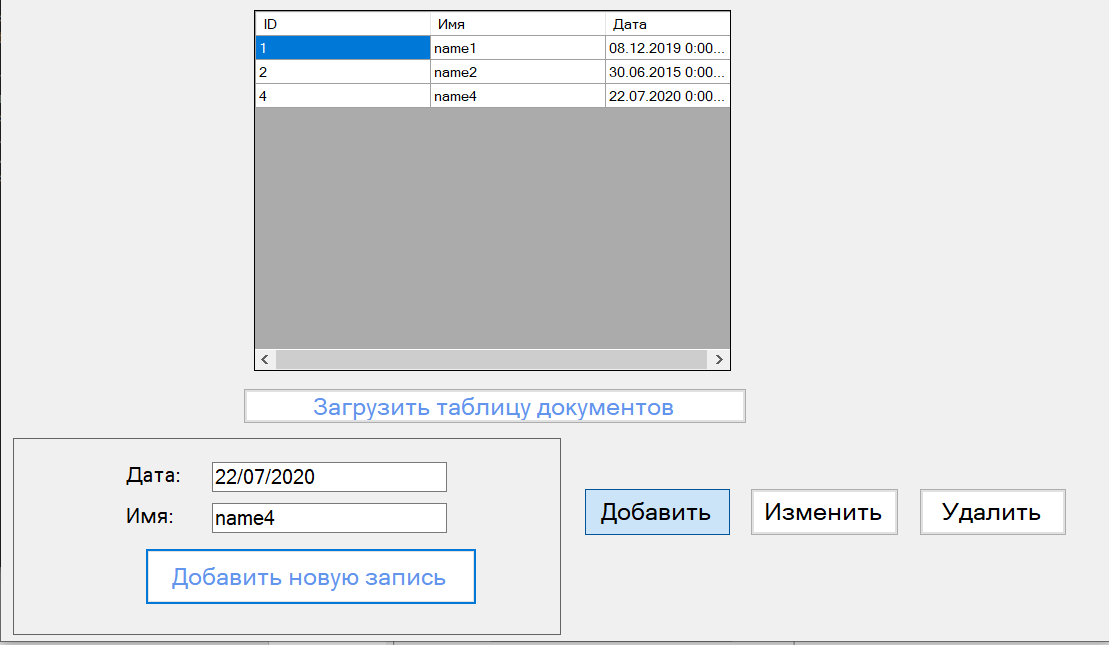


Рис. 3.1.4 – Приклад додавання інформації до бази даних

Редагування інформації бази даних:

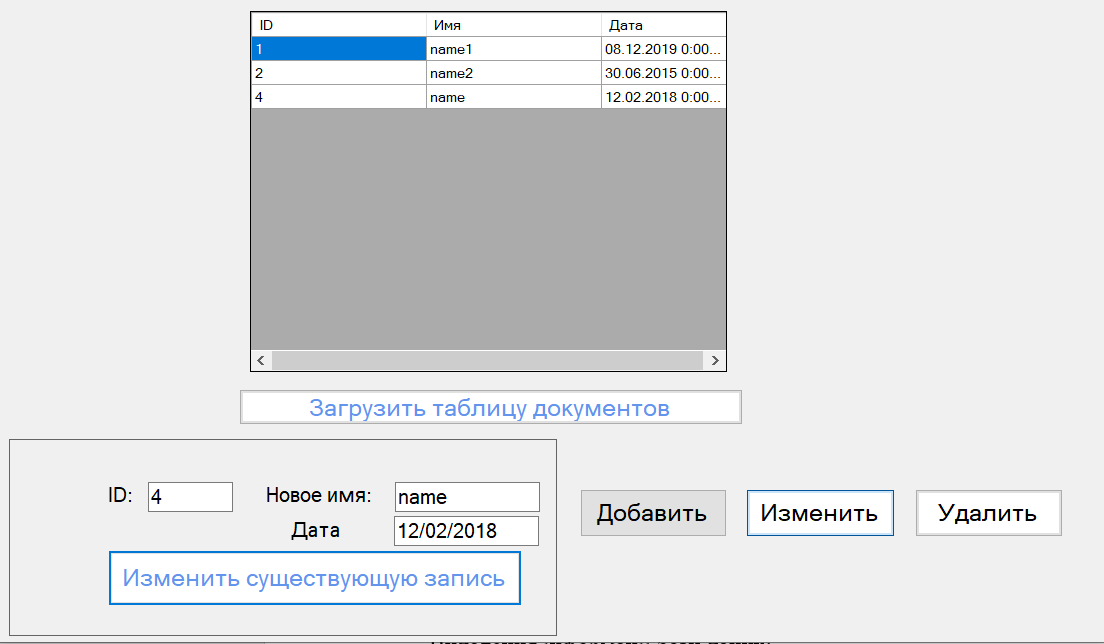


Рис. 3.1.5 – Приклад редагування інформації з бази даних

Заповнюємо поля з новим іменем та датою вже існуючого запису, вказуючи його id. Натиснувши кнопку «Изменить существующую запись», ми оновимо таблицю згідно зі змінами.

Видалення інформації бази даних:

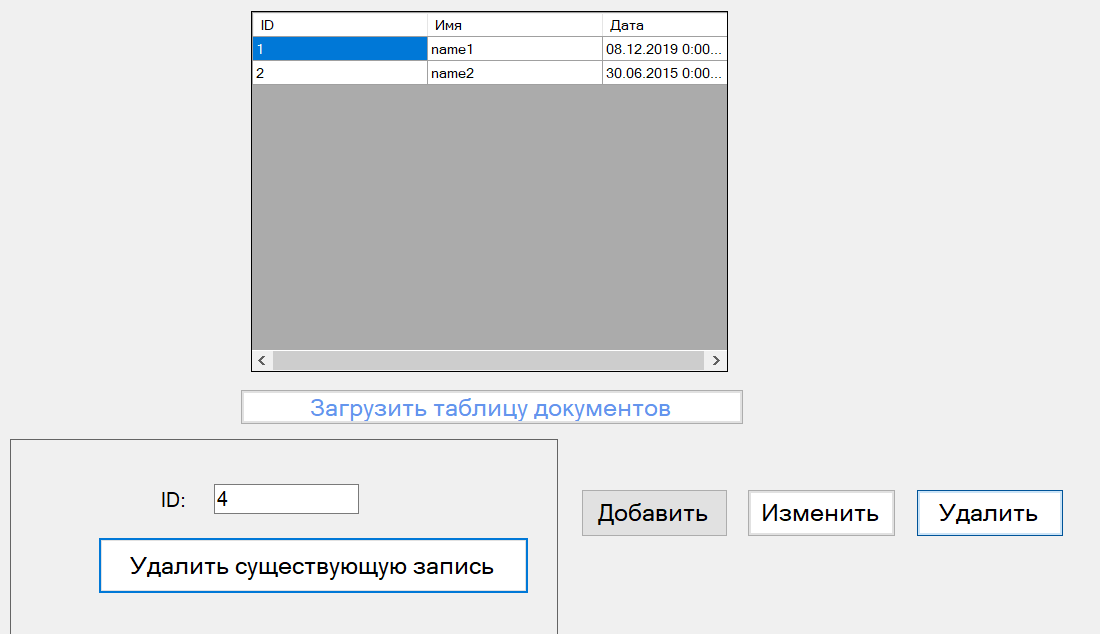


Рис. 3.1.6 – Приклад видалення інформації з бази даних

**3.2 Опис програмних класів на мові C#**

Підключення бази даних до екранної форми:

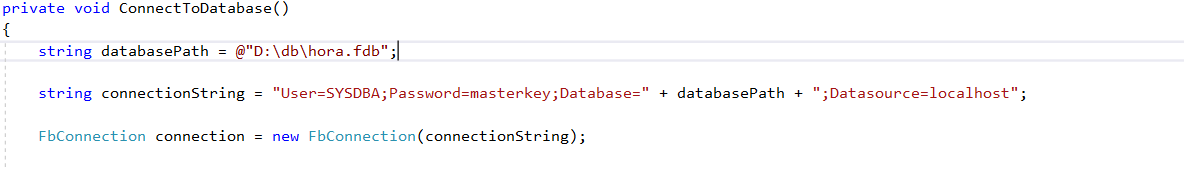


Рис. 3.2.1 – приклад відключення бази даних до екранної форми

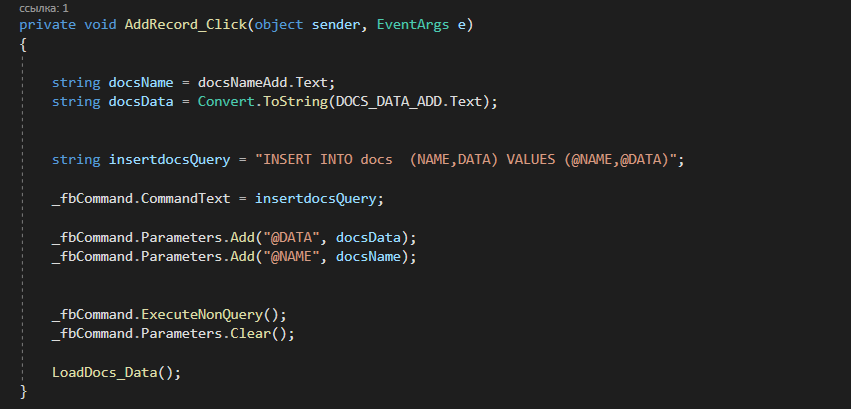


Рис. 3.2.3 – Приклад методу для додавання інформації до бази даних

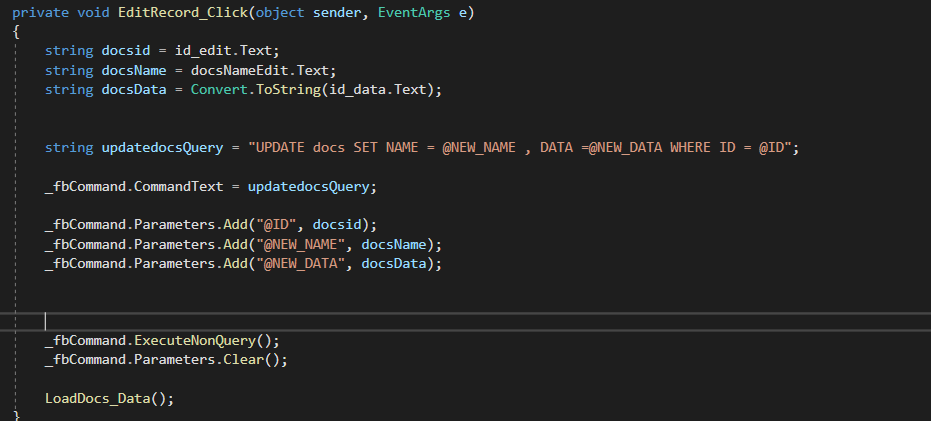


Рис 3.2.4 – Приклад методу для редагування інформації бази даних

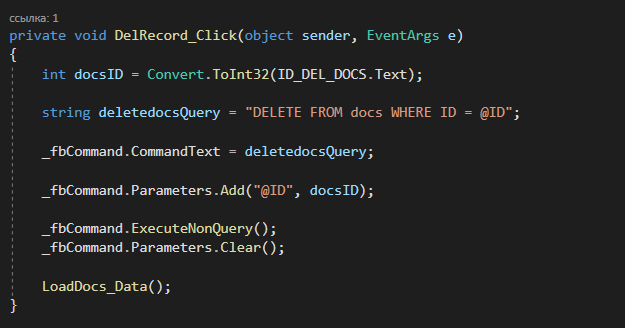


Рис. 3.2.5 – Приклад методу для видалення інформації з бази даних

**Висновки**

Виконуючи роботу з курсу «Системне програмування» за темою «Розробка програмного забезпечення «Основные средства» інформаційної системи бухгалтерії» з керуванням сутностями: внутрішня

накладна, документ, я здобула навички створення в СКБД Fiebird баз даних, та навчилась додавати, змінювати та видаляти записи з БД за допомогою SQLскриптів; створила пакети процедур з перевірками на вхідні параметри; Також зробила аналіз інтерфейсу користувача програмного модуля та розробила екранну форму для однієї з таблиць на мові С#, де створила три програмних класи, а саме:

* 1. додати екземпляр класу
  2. змінити екземпляр класу
  3. видалити екземпляр класу

У роботі наведені скріншоти роботи програми для таблиці “Документи”.