

Міністерство освіти та науки України
Чернівецький національний університет
імені Юрія Федьковича

Інформаційні системи обліку та логістика

**Методичні вказівки та завдання
до лабораторних робіт**

Чернівці
Чернівецький національний університет
2023

УДК 004.4(076.5)

Г733

Рекомендовано до друку Вченою радою факультету
математики та інформатики Чернівецького національного університету
імені Юрія Федьковича
(протокол № 6 від « 25 » січня 2023 року)

Рецензенти:

Івасюк Галина Петрівна, доцент кафедри математичного моделювання;

Фратавчан Тоня Михайлівна, доцент кафедри математичного моделювання.

Г733 Готинчан Т. І. Інформаційні системи обліку: методичні вказівки та завдання до лабораторних робіт / Т. І. Готинчан. – Чернівці : Чернівецький національний університет, 2023. – 62 с.

Видання містить методичні вказівки та завдання до лабораторних робіт, які охоплюють основні принципи створення систем обліку засобами фреймворку ВАФ.

Для студентів спеціальностей «Комп'ютерні науки» та «Системний аналіз»

УДК 004.4(076.5)

© Готинчан Т.І., 2023

© Чернівецький національний університет, 2023

Зміст

| | |
|--|----|
| Вступ..... | 4 |
| Мета і завдання навчальної дисципліни | 5 |
| Платформа VAF | 8 |
| Загальна характеристика призначень прикладних об'єктів | 8 |
| Завдання та вказівки до лабораторних робіт | 11 |
| Лабораторна робота № 1. | |
| Створення конфігурації: визначення структури..... | 11 |
| Лабораторна робота № 2 | |
| Створення конфігурації: підсистеми, форми прикладних об'єктів. Адміністрування | 23 |
| Лабораторна робота № 3 | |
| Створення конфігурації: довідники, реєстри відомостей. Документи, що є реєстраторами реєстрів відомостей..... | 31 |
| Лабораторна робота № 4 | |
| Створення конфігурації: довідники, реєстри відомостей та накопичення. Документи, що є реєстраторами реєстрів накопичення. | 42 |
| Лабораторна робота № 5 | |
| Звіти. Обробки..... | 53 |
| Література..... | 61 |

Вступ

Виникнення та розвиток обліку було викликано потребою суспільства володіти необхідною та достатньою інформацією щодо наявності та руху майна з метою контролю та управління господарськими процесами та їх результатами.

Поняття *облік* у широкому розумінні означає спостереження, вимірювання та реєстрацію певних природних і суспільних явищ. Для людей важливим є облік тих процесів, що впливають на суспільство, а об'єктом особливої уваги є господарська діяльність, основою якого становить виробництво матеріальних благ [1].

Уміння аналізувати завдання предметних галузей, які потребують обліку, виділяти сутності, зв'язки між ними, а також бізнес-процеси, допомагають студентам спеціальностей «Комп'ютерні науки» та «Системний аналіз» галузі «Інформаційні технології» у майбутньому професійно створювати додатки довільної складності

Методичні вказівки містять завдання до лабораторних робіт з курсу «Інформаційні системи обліку та логістика», а також теоретичні відомості. Передбачено, що студенти формують команди для розробки інформаційної системи для певної предметної галузі, запропонованої у переліку, тим самим набуваючи досвіду командної роботи від складання технічного завдання і розробки до тестування, збірки і використання.

У вказівках до завдань зазначені алгоритми виконання основних завдань, що є базовими для усіх команд. Кожна команда доповнює розроблену основу розробкою компонентів відповідно до свого завдання.

Мета і завдання навчальної дисципліни

Мета навчальної дисципліни полягає навчити студентів основам визначення бізнес-процесів, адміністрування та розробки прикладних застосунків у фреймворку BAF.

Дисципліна формує такі компетентності [2, 3]:

- за ОП «Комп'ютерні науки»:

Загальні компетентності:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК8. Здатність генерувати нові ідеї (креативність).

ЗК9. Здатність працювати в команді.

Фахові компетентності спеціальності:

ФК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

ФК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

ФК15. Здатність до аналізу та функціонального моделювання бізнес-процесів, побудови та практичного застосування функціональних моделей організаційно-економічних і виробничо-технічних систем, методів оцінювання ризиків їх проектування.

Наведені результати навчання за відповідною дисципліною співвідносяться із такими **програмними результатами навчання**:

ПРН1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПРН10. Використовувати інструментальні засоби розробки клієнт-серверних застосувань.

ПРН11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання).

- за ОП «Системний аналіз»:

Загальні компетентності:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК8. Здатність генерувати нові ідеї (креативність).

ЗК9. Здатність працювати в команді.

Фахові компетентності спеціальності:

ФК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

ФК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

ФК15. Здатність до аналізу та функціонального моделювання бізнес-процесів, побудови та практичного застосування функціональних моделей організаційно-економічних і виробничо-технічних систем, методів оцінювання ризиків їх проектування.

Наведені результати навчання за відповідною дисципліною співвідносяться із такими ***програмними результатами навчання:***

ПРН1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і

методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПРН10. Використовувати інструментальні засоби розробки клієнт-серверних застосунків.

ПРН11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання).

Платформа ВАФ

Загальна характеристика призначень прикладних об'єктів

Перерахування (перелічення) використовуються для опису наборів постійних значень.

Константи використовуються для роботи з постійною й умовно-постійною інформацією. Константа не може бути періодичною. Для інтерактивного редагування констант використовується форма констант, яка може створюватися за замовчуванням або призначатися розробником.

Довідники призначені для зберігання даних про безліч однотипних об'єктів, які використовуються при веденні аналітичного обліку й для заповнення документів. ВАФ підтримує роботу з багаторівневими довідниками й підлеглими довідниками. Глибина ієрархії довідників може бути довільною. Підтримуються три види підпорядкування: елементам, групам, групам і елементам. Якщо необхідно, щоб у довіднику деякі елементи існували завжди, незалежно від дій користувача, то використовують наперед визначені елементи довідника. Довідник може мати реквізити та кілька табличних частин для зберігання різних додаткових даних, що мають однакову структуру, але різні по кількості записи. Довідник немає поняття періодичних реквізитів. Для зберігання яких-небудь даних (у тому числі значень реквізитів довідників), розгорнутих у часі, тепер варто використовувати інші об'єкти конфігурації – реєстри відомостей, що підтримують періодичне збереження значень своїх ресурсів.

Документи використовуються для відображення будь-яких подій, що відбуваються на підприємстві, а також для керування розрахунками й даними. Як і довідник, документ може мати декілька табличних частин. Підтримується оперативне та неоперативне проведення документів. Є підтримка введення даних документів на підставі даних з інших об'єктів, зокрема довідників та документів.

Журнали документів призначені для перегляду документів, об'єднаних на певними критеріями для зручності пошуку.

Звіти застосовуються для одержання різноманітної інформації, яка містить підсумки або детальну інформацію, підібрану за певними критеріями.

Обробки використовуються для виконання різних сервісних або регламентних дій, наприклад, індексації цін у довідниках товарів, групове проведення документів тощо.

Регістр нагромадження (накопичення) призначений для обліку інформації про наявність і рух певних величин: матеріальних, грошових, тощо. Вони бувають реєстрами **залишків** чи **оборотів**. Формуються по виду **збільшення** (*Прибуток* +) та **зменшення** (*Видаток* –). Рухи можуть створювати різні об'єкти: документи, обробки, тощо. Ці об'єкти є реєстраторами.

Регістр відомостей призначений для опису структури зберігання даних у розрізі декількох вимірів. Регістр відомостей, що використовує прив'язку часу, називають *періодичним реєстром відомостей*. Бувають незалежними або підпорядкованими реєстратору.

План видів характеристик призначений для опису структури зберігання інформації про характеристики, створюваних користувачем.

План рахунків призначений для опису структури зберігання інформації про сукупність синтетичних рахунків підприємства, які створені для угруповання даних про його господарську діяльність.

Регістр бухгалтерії призначений для опису структури нагромадження даних, облік яких ведеться згідно з деяким планом рахунків для відображення інформації про господарські операції у бухгалтерському обліку.

План видів розрахунків призначений для опису структури зберігання інформації про можливі види розрахунків. Відмінною рисою плану видів розрахунку є те, що користувач у процесі роботи може додавати нові види розрахунку. Така можливість робить механізм періодичних розрахунків більше гнучким і дозволяє користувачеві створювати власні види розрахунку, крім тих, які задані розроблювачем як визначені.

Регістр розрахунку призначений для опису структури нагромадження даних, що є результатами різноманітних розрахунків у господарській діяльності підприємства.

Бізнес-процеси дозволяють описувати, створювати та керувати виконанням бізнес-процесів у прикладних додатках. Метою цього механізму є автоматизація ланцюжків зв'язаних операцій, направлених на досягнення загальної мети, зазвичай у контексті організаційної структури, яка визначає функціональні ролі та зв'язки.

Задача призначена для обліку завдань і описує спосіб їх розподілу по виконавцях з врахуванням організаційної структури підприємства.

Завдання та вказівки до лабораторних робіт

Лабораторна робота № 1.

Створення конфігурації: визначення структури

Постановка задачі: Створити систему ведення:

- обліку продукції, товарів та послуг;
- за своїм варіантом.

Зауваження. 1. У таблиці зазначені загальні для всіх студентів об'єкти. Для повноти вирішення завдання можна створювати додаткові об'єкти, чи реквізити у рекомендованих.

2. Надалі рекомендації надаються тільки для загальних об'єктів. Для індивідуальних об'єктів налаштування слід продумувати самим.

3. Заохочуються індивідуальні вирішення задач.

4. Для індивідуального завдання згідно із вибраним варіантом команди слід продумати додаткові об'єкти, їхню структуру та методи виконання.

Опис предметної галузі: Фірма займається виробництвом та продажом власної продукції, товарів і надання послуг. Співпрацює з постачальниками сировини, також закуповує необхідні товари. Надлишки продавати.

Фірма невелика: 10 – 15 осіб.

Торгівля: гуртова та роздрібна.

Працівники відповідних посад: кадровики, комірники, продавці, адміністратор, бухгалтер, робітники тощо.

Відповідно до опису передбачено створити прикладні об'єкти конфігурації.

| Тип об'єкта | Об'єкт |
|----------------------|--|
| Перерахування | ВидиНоменклатури Стать ТипиЦін ТипСкладу |
| Константи | ГоловБухгалтер ДатаОстанньогоЗапуску Директор КомірникЗаЗамовчуванням НазваОрганізації ОдиницяВиміруЗаЗамовчуванням СкладЗаЗамовчуванням |
| Довідники | Котрагенти Номенклатура ОдиниціВиміру Підрозділи Посади Склади Співробітники |
| Документи | ВстановленняЦінНоменклатури НадходженняТоварівПослуг ПереміщенняНоменклатури ПродажТоварівПослугПродукції |
| Журнали документів | ВстановленняЦін РухНоменклатури |
| Регістри відомостей | ЦіниНоменклатури |
| Регістри накопичення | ЗалишкиНоменклатури ПродажіКомпанії |
| Звіти | НоменклатураНаСкладі РейтингПокупців ПродажіКомпанії СпівробітникиЗаПідрозділамиПосадами |
| Обробки | Завантаження цін з Excel |

Варіанти індивідуальних завдань:

- 1) облік розрахунків через касу;
- 2) облік розрахунків через банк;
- 3) оформлення замовлення на продукцію фірми;
- 4) оформлення замовлення ТМЦ для фірми;
- 5) придбання та введення в експлуатацію ОЗ;
- 6) придбання та списання МШП;
- 7) інвентаризація наявності ТМЦ на складі. Оприбуткування надлишків;
- 8) інвентаризація наявності ТМЦ на складі. Списання недостачі;
- 9) оформлення додаткових витрат при транспортуванні товару;
- 10) формування цін з ПДВ;
- 11) передача у виробництво ТМЦ;
- 12) розрахунок собівартості продукції;
- 13) авансовий звіт підзвітної особи;
- 14) повернення ТМЦ постачальнику;
- 15) повернення продукції від клієнта;
- 16) оформлення знижок постійним клієнтам;
- 17) надходження грошей з банку/внесення готівки в банк.

Завдання:

1. Опрацювати описи предметних галузей запропонованих варіантів.
2. Сформувати команди.
3. Доповнити таблицю згідно зі своїм варіантом завдання об'єктами. Оформити опис необхідних додаткових об'єктів.
4. Оформити технічне завдання до розробки.
5. Назвати конфігурацію, визначити її властивості.
6. Створити користувача-адміністратора.
7. Створити структури об'єктів конфігурації.
8. Вивантажити файл .dt
9. Захистити роботу.
10. Завантажити звіт та архів на сторінку курсу.

Хід роботи:

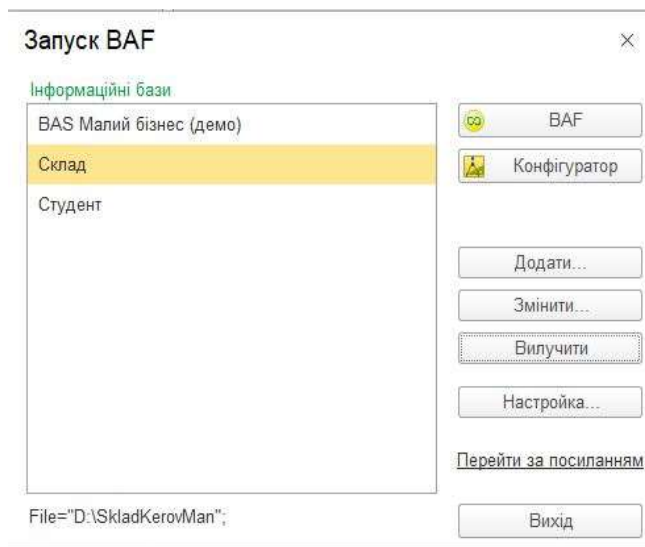
Завдання 1. Налаштування роботи

1. Створіть нову папку на будь-якому диску, наприклад, "D:\BAF\ОблікНазваКоманди".

2. Розгорніть фреймворк BAF. При встановленні передбачте українську та англійську мови. Для інтерфейсу середовища розробки встановіть англійську мову, щоб доступною була допомога з прикладами коду. Українську використовуватиме для інтерфейсу користувача.

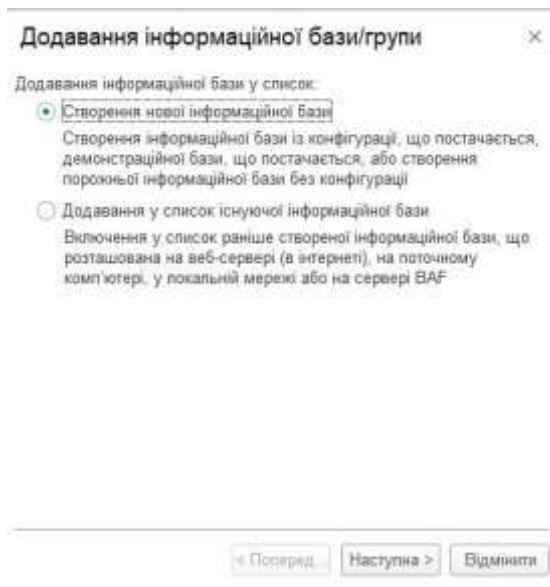
3. Завантажте фреймворк BAF.

4. Відкриється список інформаційних баз.



5. Натисніть кнопку *Додати*. Встановіть перемикач у положення *Створення нової інформаційної бази* і натисніть на кнопці *Наступна*.

6. Встановіть перемикач у положення *Створення інформаційної бази без конфігурації для розробки нової конфігурації або завантаження інформаційної бази, що була вивантажена раніше* і натисніть на кнопці *Наступна*.



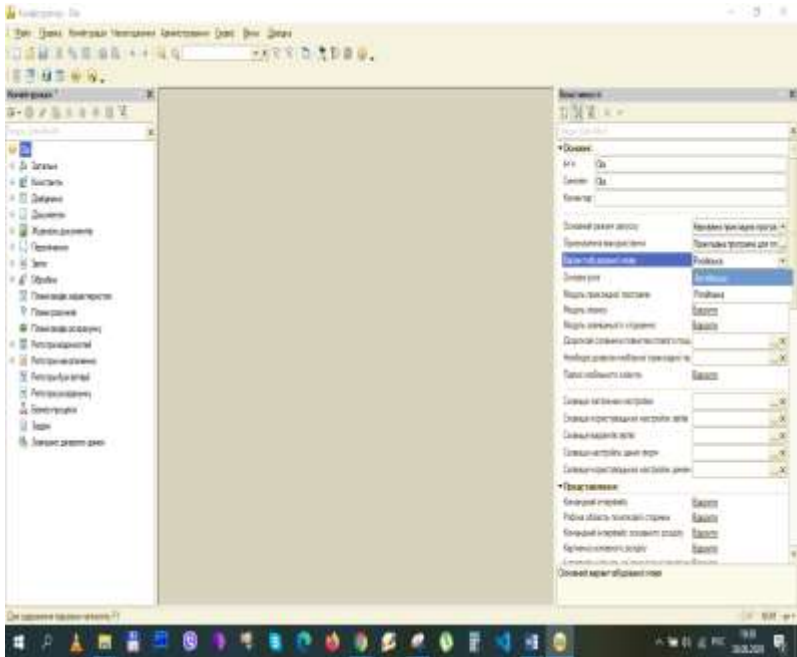
7. Вкажіть ім'я інформаційної бази, як воно відобразиться в списку баз, наприклад "Облік" і *Виберіть тип розташування інформаційної бази: на цьому комп'ютері або на комп'ютері в локальній мережі.* На наступному кроці вкажіть шлях до каталогу інформаційної бази. У цьому ж вікні оберіть мову і країну: *англійська.* Рекомендується обирати таку саму мову, як й мову інтерфейсу системи BAF.

8. Натисніть *OK.* У списку інформаційних баз з'явиться новий рядок. Для його редагування можна натиснути кнопку *Змінити.* Для вилучення інформаційної бази зі списку потрібно натиснути кнопку *Вилучити.* При цьому інформаційна база вилучається зі списку, але не вилучається фізично з диска.

9. Виберіть свою інформаційну базу. Вкажіть режим запуску *Конфігуратор.* Завантажиться *Конфігуратор.*



10. Відкрийте дерево конфігурації. Для цього використайте меню **Конфігурація => Відкрити конфігурацію**. З'явиться панель конфігурації. Панель *Конфігурація* зазвичай розташовується у лівій частині вікна програми і є деревом об'єктів конфігурації. Управляти об'єктами можна за допомогою контекстного меню або *Панелі інструментів*. Якщо конфігурація відкрита, а панелі *Конфігурація* немає на екрані, то її можна відобразити, використавши команду **Конфігурація => Вікно конфігурації**.



11. Налаштуйте властивості конфігурації. У вікні *Конфігурація*, у контекстному меню елемента *Конфігурація*, виберіть команду *Властивості*. Панель *Властивості* розташовується у правій частині екрану і містить перелік властивостей активного елемента, згрупованих по розділах: основні, дані, представлення, довідкова інформація та дія. Набір властивостей залежить від того, який об'єкт конфігурації вибраний. Наприклад,

- *Синонім*: Фірма «Прізвище» (назва конфігурації, яка відображатиметься користувачеві);
- *Основний режим запуску*: керована прикладна програма;
- *Основна мова*: англійська;
- *Розробка. Постачальник*: Ваші прізвища;
- *Версія*: № варіанта.

Завдання 2. Визначення структури проекту

1. Ознайомтесь із запропонованим зразком нетипової конфігурації.
2. Опрацюйте призначення об'єктів та логічні зв'язки між ними.
3. Доповніть дерево конфігурації згідно зі своїм варіантом завдання.
4. Визначте структури доповнених об'єктів конфігурації
5. Опишіть модель вашого проекту.
6. Зазначте хто над створенням яких об'єктів працює.
7. Оформіть звіт.

Завдання 3. Створення структур об'єктів конфігурації

• Перелічення. Константи

1. Створіть перелічення *ВидиНоменклатури*, яке міститиме значення: *Товар, Послуга, Продукція*.
2. Створіть перелічення *ТипСкладу*, яке міститиме значення: *Гуртовий, Роздрібний*.
3. Створіть перелічення *ТипиЦін* зі значеннями *Закупівельні, Гуртові та Роздрібні*.
4. Створіть перелічення *Стать* з відповідними значеннями.
5. За необхідності доповніть вище створені перелічення значеннями згідно зі своїм варіантом.
6. Перейдіть на гілку *Довідники* дерева конфігурації та створіть довідники *Співробітники, Номенклатура, ОдиниціВиміру та Склади*, задавши лише ім'я.
7. Перейдіть на гілку *Константи* дерева конфігурації та створіть константу *Директор* з властивостями:
 - Ім'я: *Директор*;
 - Тип: посилання на довідник *Співробітники*;
8. Аналогічно створіть константи *ГоловБухгалтер, КомірникЗаЗамовчуванням*.
9. Створіть константу *СкладЗаЗамовчуванням* з властивостями:
 - Ім'я: *СкладЗаЗамовчуванням*;
 - Тип: посилання на довідник *Склади*.
10. Створіть константи *ОдиницяВиміруЗаЗамовчуванням*

- Ім'я: *ОдиницяВиміруЗаЗамовчуванням*;
 - Тип: посилання на довідник *ОдиниціВиміру*.
11. Створіть константи *НазваОрганізації* (Тип: *Рядок*), *ДатаОстанньогоЗапуску* (Тип: *Дата*).
 12. Створіть константи та перелічення конфігурації свого варіанту відповідно до завдання.

- **Довідники**

1. Перейдіть на гілку конфігурації *Довідники*.
2. Відкрийте довідник *Склади*, який міститиме перелік місць зберігання номенклатури для організації:

Вкладка *Основні*:

- Ім'я: *Склади*;
- Синонім: *Склади*;

Вкладка *Ієрархія*:

- *Ієрархічний довідник* (означає, що в довіднику не можна створювати групи);

Вкладка *Дані*:

- Довжина коду: 3;
- Тип коду: *Рядок*;
- Довжина найменування: 50;
- Реквізити: *ТипСкладу* (тип: посилання на перелічення *ТипСкладу*)

Вкладка *Нумерація*:

- Автонумерація: Так.

3. Аналогічно оформіть довідники *ОдиниціВиміру*, *Посади*, *Підрозділи*, але не створюючи реквізити.
4. Відкрийте довідник *Співробітники* з реквізитами *Прізвище*, *Імя* (синонім *Ім'я*), *ПоБатькові* (синонім *По батькові*) - (Тип: *Рядок*), *Посада* (тип: посилання на довідник *Посади*), *Підрозділ* (тип: посилання на довідник *Підрозділи*), *ТабельнийНомер* (Тип: *Рядок*). *ДатаНародження* (Тип: *Дата*). Цей довідник також неієрархічний.
5. Створіть довідник *Контрагенти* з реквізитами *ПовнаНазва*, *Адреса* та табличною частиною *КонтактнаІнформація* з реквізитами *ІмяКонтОсоби* (синонім *Ім'я контактної особи*), *Посада*, *Телефон*, *email*. **Типи даних за змістом.** До-

відник має бути ієрархічним (вид ієрархії: *Ієрархія груп і елементів*).

6. Створіть довідник *Номенклатура*, який міститиме назви товарів, продукції та послуг. Додайте у довідник *Номенклатура* такі реквізити *ЦінаЗакупівлі*, *ЦінаРоздрібна*, *ЦінаГуртова* - (Тип: *Число*), *ОдиницяВиміру* (Тип: посилання на довідник *ОдиниціВиміру*), *ВидНоменклатури* (Тип: посилання на перелічення *ВидиНоменклатури*), *ДатаПерегляду*. Зробіть довідник *Номенклатура* ієрархічним (вид ієрархії: *Ієрархія груп і елементів*). **Слідкуйте за необхідним представленням типів даних.**

7. Створіть довідники конфігурації свого варіанту відповідно до завдання.

- **Документи**

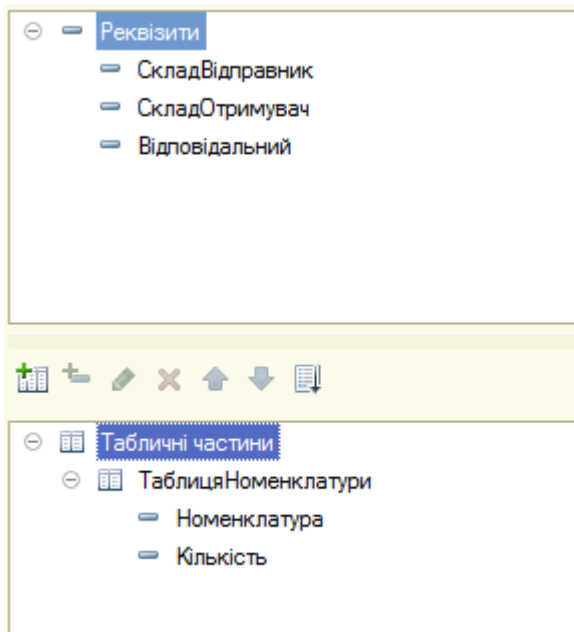
1. Перейдіть на гілку конфігурації *Документи*.

2. Створіть документ *ВстановленняЦінНоменклатури* з табличною частиною *Номенклатура* (реквізити: *НоменклатураНазва*, *ТипЦіни*, *Ціна*) та реквізитами *ВибірВидНоменклатури* (тип: посилання на перелічення *ВидиНоменклатури*), *Бухгалтер* (Тип: посилання на довідник *Співробітники*). **Типи даних за змістом.**

3. Створіть документ *НадходженняТоварівПослуг*. У реквізити документа додайте *Склад*, *Контрагент*, *Комірник*, *Сума* а у табличну частину *ТаблицяНоменклатури*: *Номенклатура*, *ОдиницяВиміру*, *Кількість*, *Ціна* та *Сума*. **Типи даних за змістом. Слідкуйте за представленням типів даних.**

4. Скопіюйте документ *НадходженняТоварівПослуг* і переименуйте копію у *ПрожажТоварівПослугПродукції*. Зверніть увагу, що всі реквізити і табличні частини скопіювалися разом з документом. Переименуйте його на *Продаж товарів, послуг, продукції*.

5. Створіть документ *ПереміщенняНоменклатури*, за допомогою якого можна переміщувати товари з одного складу до іншого.



6. Створіть документи конфігурації свого варіанту відповідно до завдання.

- **Регістри відомостей та накопичення**

1. Перейдіть на гілку конфігурації *Регістри відомостей*.

2. Створіть реєстр відомостей *ЦіниНоменклатури*, що зберігатиме інформацію по цінам номенклатури за типами цін. *Ціна* (Тип: число) є ресурсом, а *Номенклатура* (тип: посилання на довідник *Номенклатура*), *ТипЦіни* (тип: посилання на перелічення *ТипиЦін*) – вимірами. **Виміри, які використовуються у пошуку чи відборі, роблять ведучими або індексують.** Задайте властивість вимірам «ведуче». Для кожної позиції товару треба зберігати значення кожної із цін. Ціна товару не повинна змінюватися більше одного разу на день.

3. Режим запуску: підпорядкування реєстратору. Реєстратором призначте документ *ВстановленняЦінНоменклатури*.

4. Перейдіть на гілку конфігурації *Регістри накопичення*.

5. Створіть реєстр накопичення *ЗалишкиТоварів* (вид: *залишки*). У реєстрі додайте виміри *Номенклатура* та *Склад*, ресурс *Кількість*. Установіть на вкладці *Реєстратори* навпроти документів *НадходженняТоварівПослуг* і *ПродажТоварівПослугПродукції, ПереміщенняНоменклатури*.

6. Додайте у конфігурацію реєстр накопичення *ПродажіКомпанії* Встановіть для цього реєстру вид *обороти*. Сформуйте структуру реєстру таким чином, щоб він зберігав інформацію про те, якому контрагентові, коли і на яку суму був проданий товар. Призначте реєстратор.

7. Створіть реєстри конфігурації свого варіанту відповідно до завдання.

Завдання 4. Звітність

1. Вивантажте інформаційну базу у файл з розширенням *.dt*, який перейменуйте й у назву до прізвищ додайте номер лабораторної роботи.

2. Завантажте файл та звіт з переліком виконавців та виконаних видів робіт на електронний курс.

Запитання для повторення

1. Що доцільно оформлювати як константи та перелічення?
2. Які групи реквізитів можна задати у довіднику та документах?
3. Які види ієрархії можна визначати для довідників?
4. Які реквізити за замовчуванням є у довідника, документа, журналу документів?
5. Як підпорядкувати реєстр документу?
6. Чи бувають реєстри накопичення незалежними?
7. Види реєстрів відомостей.

Лабораторна робота № 2

Створення конфігурації: підсистеми, форми прикладних об'єктів. Адміністрування

Завдання:

1. Створити необхідні форми прикладних об'єктів.
2. Описати ролі користувачів.
3. Задати підсистеми і об'єкти до них.
4. Створити командний інтерфейс з прив'язкою ролей.
5. Створити користувачів.
6. Прописати параметри запуску системи.
7. Вивантажити файл .dt
8. Захистити роботу.
9. Завантажити звіт та архів на сторінку курсу

Хід роботи:

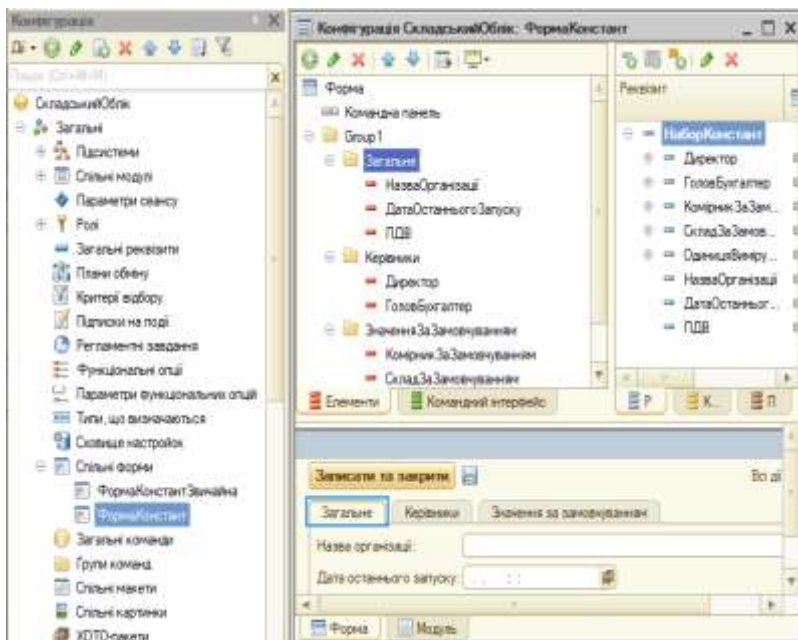
Завдання 1. Форми об'єктів конфігурації

1. Зайдіть на гілку конфігурації *Загальні* та додайте **спільну керовану форму**, яку назвіть *ФормиКонстант*. Додайте на неї усі константи.

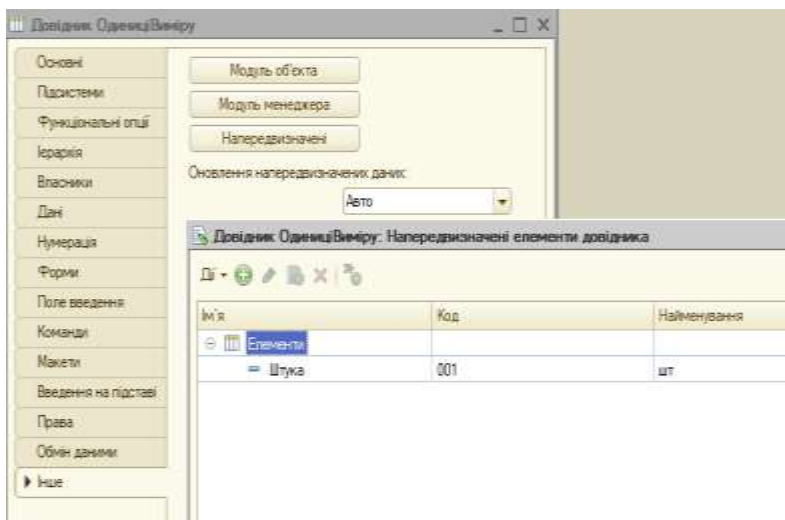
2. Згрупуйте константи по вкладках. Для цього до елементів додайте через контекстне меню групу-сторінки, а всередині групи – 3 сторінки *Загальне*, *Керівники*, *ЗначенняЗаЗамовчуванням*.

Зуваження. У залежності від варіанту у вас можуть бути й інші константи, які не зображені на малюнку.

3. На формі констант для елемента керування *ДатаОстанньогоЗапуску* встановіть властивість *Лише перегляд*.

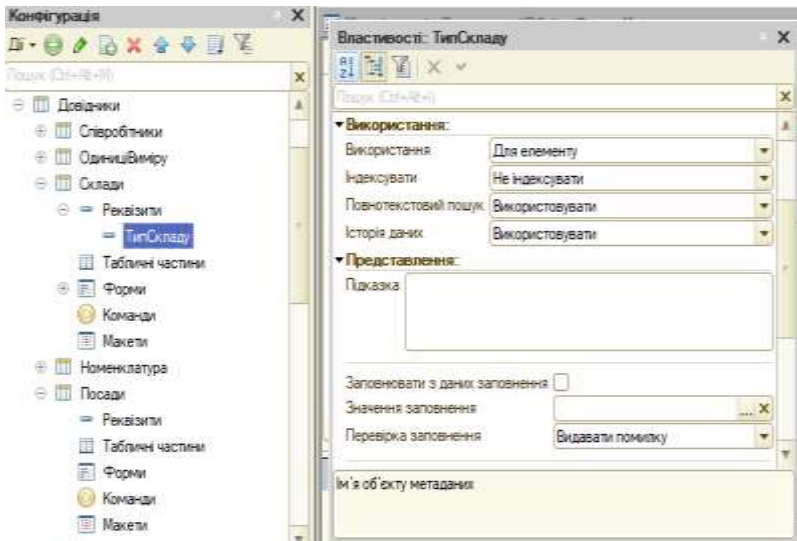


4. Для перелічень створювати форми не потрібно.
5. Перейдіть на гілку *Довідники*.
6. Для довідників *Посади*, *Підрозділи*, *Одиниці вимірювання* створювати форми не потрібно. Форми генеруватимуться системою самі й здійснюватиметься перевірка заповнення реквізитів за замовчуванням.
7. Для довідника *Одиниці вимірювання* створіть напередвизначений елемент *штука*.



8. Створіть форму елемента довідника Склади.

9. Елементу форми *Тип складу* задайте властивість *АвтоВідміткиЗаповнення*, а реквізиту *ТипСкладу* властивість перевірки заповнення



10. Створіть форми елементів та списків решти довідників. За необхідності, додавши вкладки або решта елементів керування. Додайте ознаки обов'язковості заповнення та перевірки заповнення реквізитів. **Зауваження.** У формі списку вивести не всі реквізити, а які ідентифікують довідник.

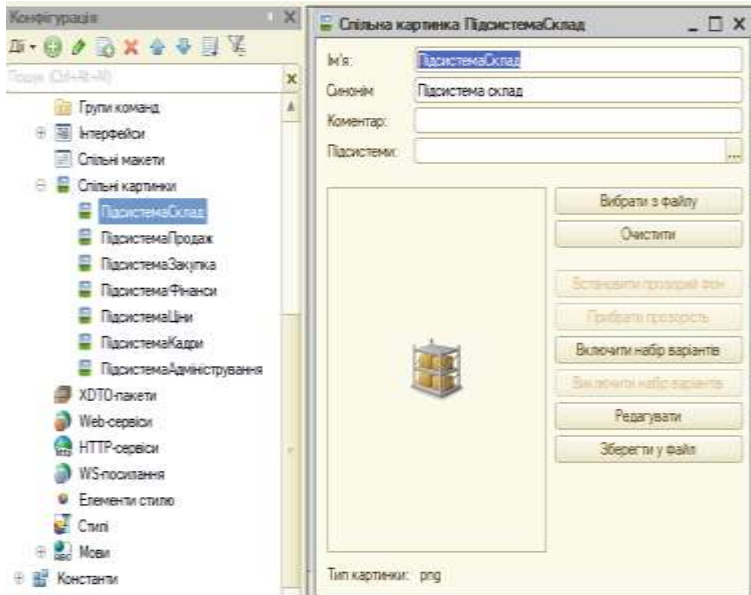
11. Перейдіть на гілку *Документи*.

12. Створіть форми списку та документа для документів *Встановлення цін номенклатури* та *Надходження товарів і послуг*. На формі документів для них додайте напис-декорацію *Назва організації*.

Зауваження. У формі списку вивести не всі реквізити, а які ідентифікують документ.

Завдання 2. Підсистеми. Ролі

1. Перейти на гілку *Загальні*.
2. Завантажити картинку, яка б відображала підсистему *Закупка, Склад, Продаж, Адміністрація* та за вашим варіантом.



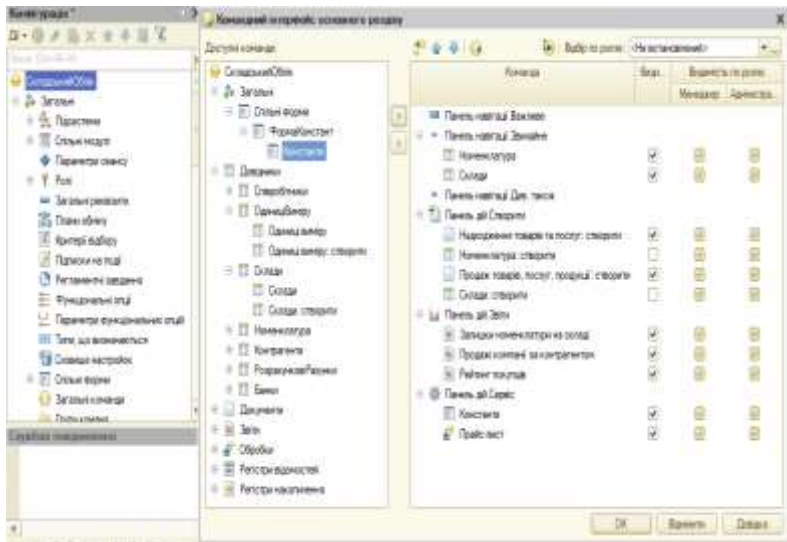
3. Створити верхній рівень підсистем *Закупка, Склад, Продаж, Адміністрація*. За потреби додати ще підсистеми. Наповнити їх відповідними об'єктами конфігурації.

4. До кожної підсистеми продумати внутрішню структуру і створити її.

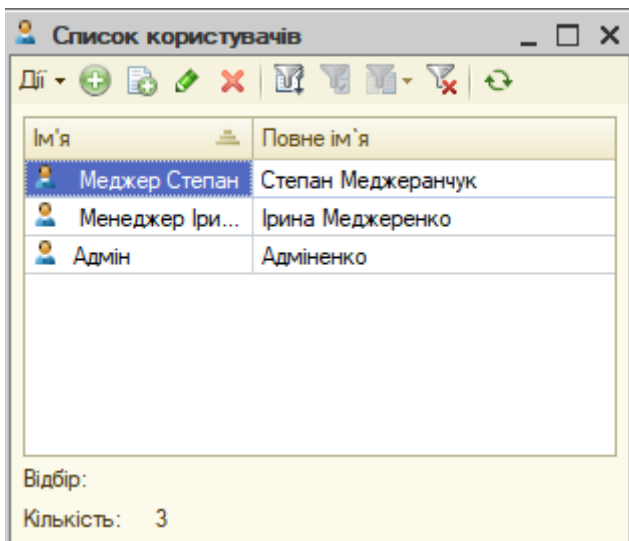
5. Створити ролі Адмін, Менеджер. Наповнити їх відповідними правами.

6. Відкрити командний інтерфейс на кожному рівні підсистем і задати відображення об'єктів відповідно до ролей.

7. Відкрити інтерфейс основного розділу і наповнити його об'єктами, що найчастіше використовуються.

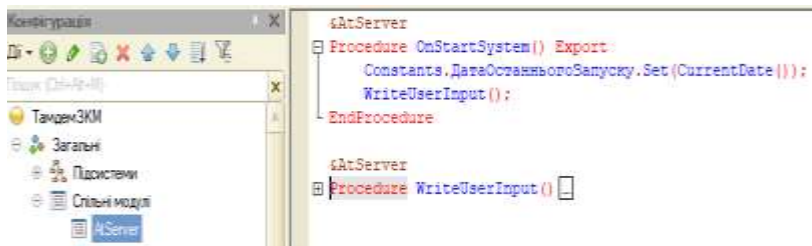


8. Створити користувачів з відповідними ролями. Для цього Зайдіть на вкладку *Адміністрування* та виберіть команду *Користувачі*

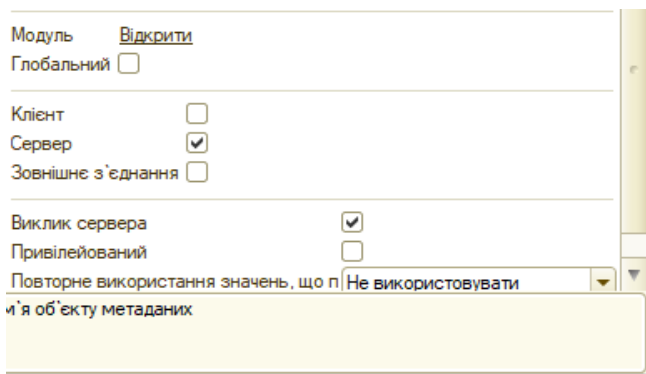


Завдання 3. Запуск системи у режимі налагодження

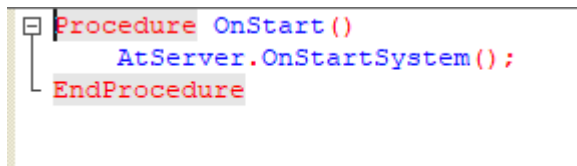
1. Створити модуль спільний модуль *AtServer* у залежності від вибору мови програмування. У середині набрати процедуру, текст якої наведено на малюнку



2. Для цього спільного модуля задати властивості



3. Відкрити модуль керованої прикладної програми конфігурації і набрати процедуру відповідно до вибраної мови програмування



4. Запустити систему у режимі налагодження як адміністратор. Заповнити довідники елементами, які необхідно задати у константах. Перевірити заповнюваність елементів форми об'єктів бази.

5. Заповнити константи. Переконайтесь, що константа *Дата останнього запуску* заповнюється системою, а не вручну і редагуванню не піддається.

Завдання 4. Звітність

3. Завантажте інформаційну базу у файл з розширенням .dt, який перейменуйте й у назву до команди додайте номер лабораторної роботи.

4. Завантажте файл .dt та звіт з переліком виконавців та виконаних видів робіт на сторінку курсу.

Запитання для повторення

1. Які види модулів підтримує фреймворк?
2. Які є види директив?
3. Яка директива є за замовчуванням?
4. Скільки форм можуть мати прикладні об'єкти?
5. Як налаштувати перевірку заповнення даних у комірках?
6. Що таке керований інтерфейс і як його налаштувати?
7. Як налаштувати права користувачів?

Лабораторна робота № 3

Створення конфігурації: довідники, реєстри відомостей. Документи, що є реєстраторами реєстрів відомостей

Завдання:

1. Створити необхідні форми прикладних об'єктів.
2. Описати ролі користувачів.
3. Задати підсистеми і об'єкти до них.
4. Створити командний інтерфейс з прив'язкою ролей.
5. Створити користувачів.
6. Прописати параметри запуску системи.
7. Вивантажити файл .dt
8. Захистити роботу.
9. Завантажити звіт та архів на сторінку курсу

Хід роботи:

Завдання 1. Реєстри відомостей і документи, що є реєстраторами.

1. Відкрийте конструктор документа *ВстановленняЦін-Номенклатури*.

2. За допомогою конструктора рухів сформуєте процедуру проведення документа. Процедура відобразиться у модулі об'єкта, У процедурі у модулі об'єкта відобразиться код, за яким заповнюється реєстр відомостей *ЦіниНоменклатури*.

//=====

Procedure Posting(Cancel, Mode)

// Даний фрагмент побудований конструктором.

```
// При повторному використанні конструктора, внесені вручну зміни будуть втрачені!!!
```

```
// реєстр ЦіниНоменклатури
RegisterRecords.ЦіниНоменклатури.Write = True;
For Each CurRowНоменклатура In Номенклатура Do
    Record = RegisterRecords.ЦіниНоменклатури.Add();
    Record.Period = Date;
    Record.Номенклатура = CurRowНоменклатура.НоменклатураНазва;
    Record.ТипЦіни = CurRowНоменклатура.ТипЦіни;
    Record.Ціна = CurRowНоменклатура.Ціна;
EndDo;

EndProcedure
```

```
//=====
```

3. Для форми документа допишіть обробку події *При відкритті (OnOpen)*, у якій за замовчуванням заповнюється назва організації та бухгалтер.

```
//=====
```

```
&AtServer
Function GetOrganizationName()
return Constants.НазваОрганізації.Get();
EndFunction
```

```
//=====
```

```
&AtServer
Procedure SetOrganizationName()
    orgName = Organization.GetOrganizationName();
    ThisForm.ChildItems.Organization.Title = "Назва організації : " + orgName;
EndProcedure
```

```
//=====
```



```
&AtC&AtClient  
Procedure OnOpen(Cancel)  
SetOrganizationName();  
EndProcedure
```

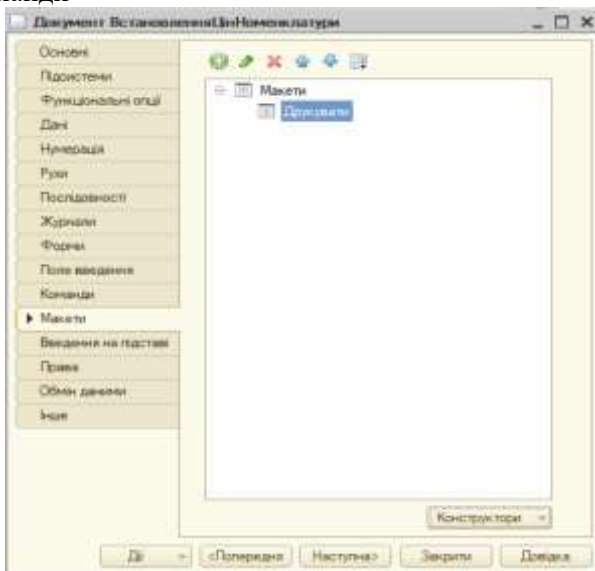
//=====

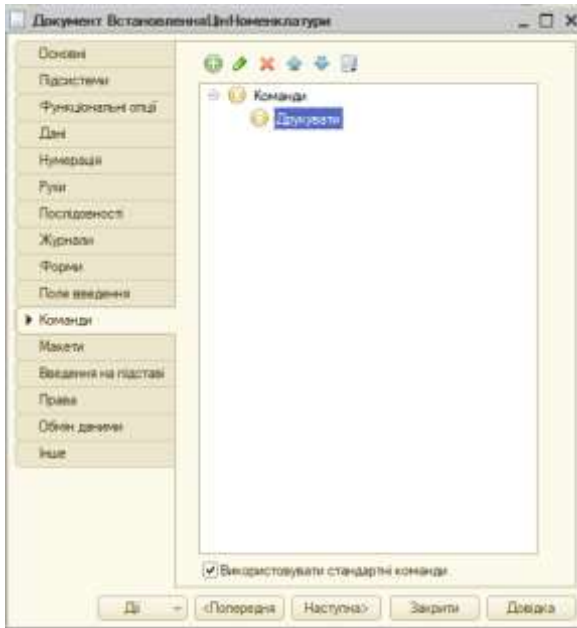
Зауваження. Оскільки назва організації має відобразитись у кожному документі, то доцільно написати відповідну процедуру у спільному модулі. У цій процедурі, яка виконується на сервері, має отримуватись значення із відповідної константи. Цю процедуру викликати у формі документа.

4. Дopiшіть процедури заповнення елемента форми Бухгалтер даними за замовчуванням з відповідної константи. При цьому зробіть перевірку на заповненість поля.

5. Зробіть перевірку заповнення обов'язкових елементів табличної частини та заголовку.

6. Створіть друковану форму документа. Для цього у конструкторі документа зайдіть на вкладку Макети і запустіть конструктор друку. При цьому виберіть які елементи будуть у заголовку та тілі та підвалі табличного документа. Згенерується макет, команда і код команди





Процедури команди:

//-----

&AtClient

Procedure CommandProcessing(CommandParameter,
CommandExecuteParameters)

//{{_PRINT_WIZARD(Друквати)

Spreadsheet = New SpreadsheetDocument;

Друквати(Spreadsheet, CommandParameter);

Spreadsheet.ShowGrid = False;

Spreadsheet.Protection = True;

Spreadsheet.ReadOnly = True;

Spreadsheet.ShowHeaders = False;

Spreadsheet.Show();

//}}

EndProcedure

&AtServer

Procedure Print(Spreadsheet, CommandParameter)

```
Documents.ВстановленняЦінНоменклатури.Друкувати(Spreadsheet, CommandParameter);  
EndProcedure
```

```
//=====
```

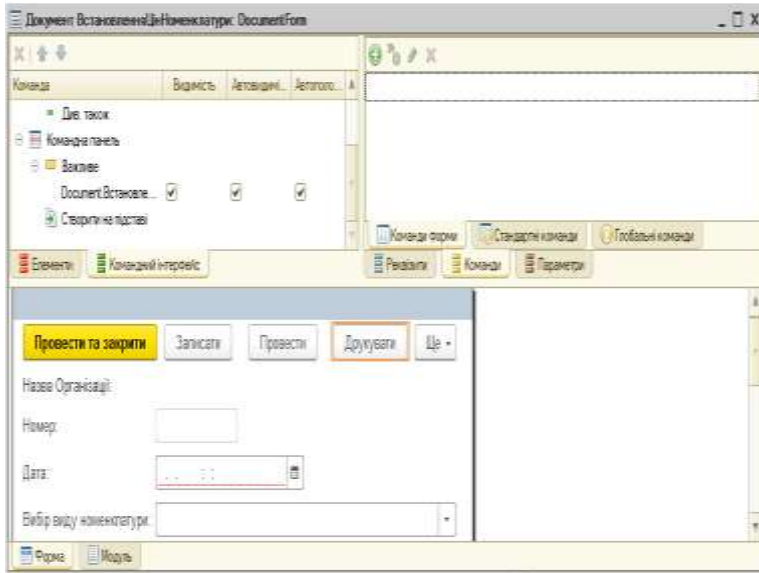
Процедура менеджера

```
//=====
```

```
Procedure Print(Spreadsheet, Ref) Export  
    //{{_PRINT_WIZARD(Друкувати)  
    Template =  
    Documents.ВстановленняЦінНоменклатури.GetTemplate("Друкувати");  
    Query = New Query;  
    Query.Text =  
    "SELECT  
    |         ВстановленняЦінНоменклатури.Date,  
    |         ВстановленняЦінНоменклатури.Number,  
    |         ВстановленняЦінНоменклатури.Бухгалтер,  
    |         ВстановленняЦінНоменклатури.ВибірВидНоменклатури,  
    |         ВстановленняЦінНоменклатури.Номенклатура.(  
    |             LineNumber,  
    |             НоменклатураНазва,  
    |             ТипЦіни,  
    |             Ціна  
    |         )  
    |FROM  
    |         Document.ВстановленняЦінНоменклатури  
AS ВстановленняЦінНоменклатури  
    |WHERE  
    |         ВстановленняЦінНоменклатури.Ref IN  
(&Ref);  
    Query.Parameters.Insert("Ref", Ref);  
    Selection = Query.Execute().Select();  
    .....  
EndProcedure
```

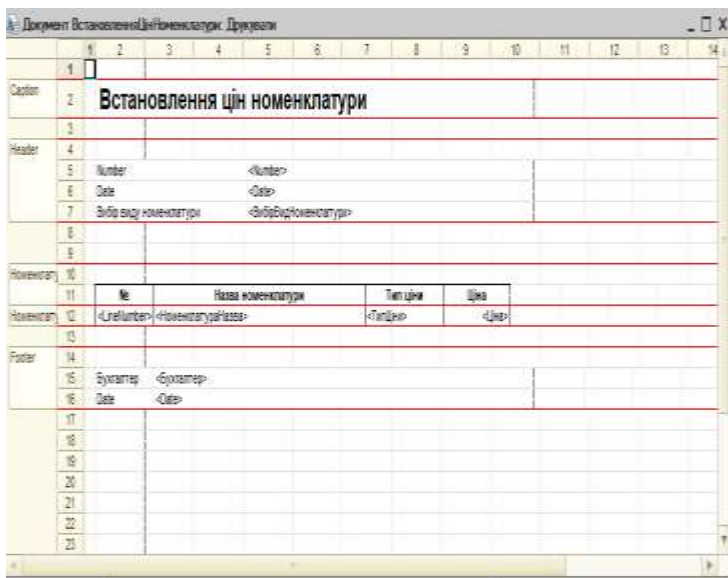
//=====

10. Для відображення кнопки команди у формі документа переконайтеся, що проставлені прапорці



11. Для доступу до команди користувачем у ролі не забудьте надати права на перегляд команди.

12. Згенерований макет удоскональте до вимог шаблону.



Завдання 2. Довідники

1. Відкрийте форму довідника *Співробітники*. У модулі форми створіть процедури відповідно вибраної мови програмування:

```
//=====
&AtServer
Procedure GetFullName()
    Object.Description= Object.Прізвище+" "+
    Left(Object.Імя,1) + "." +Left(Object.ПоБатькові,1) + ".";
EndProcedure

&AtClient
Procedure FullNameOnChange(Item)
    // Вставити вміст обробника
    GetFullName();
EndProcedure

//=====
```

Для цього виберіть поле вводу Прізвище і через властивості задайте процедуру *При змінні (OnChange)* як на сервері, так і на клієнті. Замініть у назві процедури слово Прізвище на ПІБ і додайте процедуру на клієнті до решти двох полів, які в тілі процедури зазначені.

При цьому полю вводу Найменування (*Description*) (не реквізиту!) задайте властивість не тільки читання та змініть напис на ПІБ.

Зауваження. Вітається назви процедур і функцій, а також змінних указувати мовою вибраної ВМП.

2. Відкрийте довідник *Номенклатура*. Зробіть реквізити *Вид-Номенклатури* й *ОдиниціВиміру* обов'язковими для заповнення. Задайте текст повідомлення помилки або залиште за замовчуванням.

2.1. У модулі форми елемента пропишіть процедури обробки подій

```
//=====
&AtClient
Procedure OnOpen(Cancel)
    Object.ДатаПерегляду = CurrentDate();
    ЗаповненняЦін();
EndProcedure
//=====
```

2.2. Доповніть попередню процедуру заповненням одиниці виміру значенням з константи за замовчуванням (Для цього згадайте як це робилось у попередній роботі).

2.3. У модулі форми елемента створіть процедуру користувача *ЗаповненняЦін()* (*FillPrices()*) для елемента форми Дата перегляду.

```
//=====
```

```

&AtServer
  Procedure FillPrices()
    // У цьому місці запустить через контексне меню конструктор
    // запитів із обробкою результату
    Query = New Query(" тут має бути тіло запиту, де вибираються дані
    // останніх цін за періодом та типом цін для заданої номенклатури.
    // Тобто реквізити ДатаПерегляду і Номенклатура мають бути параметрами.
    // Використовуємо віртуальну таблицю зріз останніх для реєстру відомостей
    // Ціни номенклатури
    |Select
    |      ");
    // Тут мають бути описані параметри
    // Продовження тіла процедури
    Result = Query.Execute();
    If Not Result.isEmpty() Then
      Selection = Result.Select();
      While Selection.Next() Do
        If Selection.ТипЦіни = Enums.ТипиЦін.Гуртова Then
          Object.ЦінаГуртова = Selection.Ціна;
        ElseIf Selection.ТипЦіни = Enums.ТипиЦін.Закупівельна
Then
Object.ЦінаЗакупівлі = Selection.Ціна;
        ElseIf Selection.ТипЦіни = Enums.ТипиЦін.Роздрібна Then
          Object.ЦінаРоздрібна = Selection.Ціна;
        EndIf;
      EndDo;
    Else

```

```

Object.ЦінаГуртова =0;
Object.ЦінаЗакупівлі =0;
Object.ЦінаРоздрібна =0;

EndIf;
EndProcedure

```

```
//=====
```

І викличте її у процедурі при змінні дати перегляду

```
//=====
```

```
&AtClient
```

```

Procedure ДатаПереглядуOnChange(Item)
    FillPrices();

```

```
EndProcedure
```

```
//=====
```

3. Для елемента номенклатури створіть друковану форму.
4. Додаткове завдання*. Створіть Форму групи для довідника Номенклатура. Вибирайте не всі реквізити, а лише ті, які ідентифікують номенклатуру. Створіть друковану форму списку номенклатури.
5. Заповніть модулі форм решти об'єктів відповідно до завдання.
6. Доповніть об'єкти конфігурації свого варіанту відповідно до завдання.

Завдання 3. Заповнення інформаційної бази.

1. Збережіть конфігурацію.
2. Перейдіть у режим налагодження системи та заповніть інформаційну базу.
4. За допомогою *Синтаксис-Помічника* ознайомитись із синтаксисом і призначенням функцій, процедур, методів, які використовувались.

Завдання 4. Звітність

1. Вивантажте інформаційну базу у файл з розширенням .dt, який перейменуйте й у назву до прізвища додайте номер лабораторної роботи.

2. Завантажте файл та звіт з переліком виконавців та виконаних видів робіт.

Запитання для повторення

1. Як створити запит до об'єкта?
2. Як створити друковану форму об'єкта?
3. Як побудувати розшифрування комірки макету?
4. Основні розділи макету.
5. Як оформити виклик події?
6. Як організувати вивід даних з у форму об'єкта?
7. Як організувати вивід результату запиту в табличну частину об'єкта?

Лабораторна робота № 4

Створення конфігурації: довідники, реєстри відомостей та накопичення. Документи, що є реєстраторами реєстрів накопичення.

Завдання:

1. Створити документи, реєстри накопичення, звіти та обробки, зазначені у таблиці та відповідно до свого варіанту.
2. Прописати модулі форм та об'єктів.
3. Відредагувати інтерфейси користувачів.
4. Перевірити роботу створених об'єктів, перейшовши у режим налагодження роботи
5. Заповнити інформаційну базу.
6. Вивантажити файл .dt
7. Захистити роботу.
8. Завантажити звіт та архів на сторінку курсу.

Хід роботи:

Завдання 1. Документи та журнали документів.

1. Перейдіть на гілку конфігурації *Документи*.
2. Відкрийте документ *НадходженняТоварівПослуг*.
 - 2.1. Відкрийте форму документа
 - 2.2. Встановіть для *Табличного поля* форми *Підвал*, *Заголовок*. Виділіть стовпчик *Сума* і встановіть для нього властивість *Відображати у підвалі* та пропишіть шлях до даних.
 - 2.3. Щоб здійснити перерахунок даних у табличній частині при зміні ціни та кількості товарів зробіть наступне:
 - у формі документа *НадходженняТоварівПослуг* виділіть поле введення *Кількість*;
 - у обробці події *OnChange* напишіть код:

//=====

```

&AtClient
Procedure CountOnChange (Item)
    currentRow =
Items.ТаблицяНоменклатури.CurrentData;
    currentRow.Сума = currentRow.Ціна *
currentRow.Кількість;
EndProcedure
//=====

```

2.4. Змініть назву процедури так, щоб вона відображала пере-
рахунок не лише при змінні кількості, а й ціни. Додайте цю саму
обробку події для поля введення *Ціна*.

2.5. Для поля введення номенклатура у обробці події *OnChange*
напишіть код, за яким би підтягувалась одиниця вимірювання но-
менклатури.

```

//=====
&AtServer
Function GetUnit (Номенклатура)
    return Номенклатура.ОдиницяВиміру;
EndFunction

&AtClient
Procedure НоменклатураOnChange (Item)
    currentRow =
Items.ТаблицяНоменклатури.CurrentData;
    currentRow.ОдиницяВиміру =
GetUnit (currentRow.Номенклатура);
EndProcedure
//=====

```

2.6. Для таблиці документа напишіть обробку події *OnEditEnd()*;

```

//=====
&AtClient
Procedure ТаблицяНоменклатуриOnEditEnd(Item,
NewRow, CancelEdit)
    // Вставити вміст обробника.
    Object.Сума =
Object.ТаблицяНоменклатури.Total("Сума");
EndProcedure
//=====

```

Зауваження. Оскільки сумарні дані поля Сума відображаються у підвалі стовпчика, то доцільно це поле не відображати на формі, а використовувати як дані для відображення у графах журналів.

2.7. Оскільки при придбанні використовують закупівельні ціни, то для поля *Номенклатура* таблиці документа напишіть обробку події *OnChanhe*, яка автоматично вставлятиме у табличну частину документа ціни вибраного найменування.

```

//=====
&AtServer
Function GetPrice(Номенклатура)
    priceType = Enums.ТипЦін.Закупівельна;
    query = new Structure("Номенклатура", Номен-
клатура);
    query.Insert("ТипЦін", priceType);

    purchasePrice =
InformationRegisters.ЦіниНоменклатури.GetLast(Object.Date,query);    return purchasePrice.Ціна;
EndFunction

```

```

&AtClient
Procedure НоменклатураOnChange(Item)

```

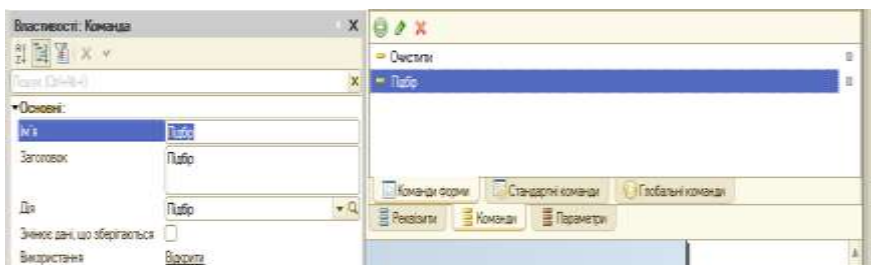
```

        currentRow =
Items.ТаблицяНоменклатури.CurrentData;
        price = GetPrice(currentRow.Номенклатура);
        // Дописати код так, щоб одночасно додавалась
        // одиниця вимірювання, кількість =1 та підрахову-
        // валась сума
EndProcedure

//=====

```

- 2.8. Створіть кнопку *Підбір* і напишіть обробку події
- спочатку створіть стандартну команду *Підбір*
 - напишіть обробку події



```

//=====
&AtClient
Procedure Підбір(Command)
    formOwner = Items.ТаблицяНоменклатури;
    formParams = New Structure;
    formParams.Insert("ChoiceMode", true);

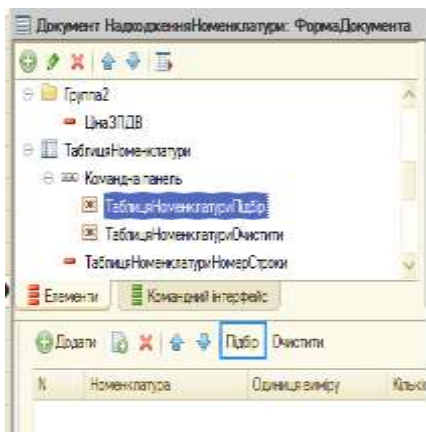
    formParams.Insert("CloseOnChoice", false);
    formParams.Insert("ChoiceFoldersAndItems"
, FoldersAndItemsUse.Items);

```

```

OpenForm("Catalog.Номенклатура.ChoiceForm",
formParams, FormOwner);
EndProcedure
//=====

```



- на командній панелі табличної частини задайте кнопку *Підбір*

Аналогічно створіть кнопку *Очисти* для очистки табличної частини.

2.9. Для табличної частини *ТаблицяНоменклатури* напишіть обробку події *Обробка вибору ChoiceProcessing* з перевіркою повторення вибраних найменувань.

```

//=====
&AtClient
Procedure
    ТаблицяНоменклатуриChoiceProcessing(Item, SelectedValue,
StandardProcessing)
    Selected = 0;
    For Each Row In Object.ТаблицяНоменклатури do
    if Row.Номенклатура = SelectedValue
    then
    Selected = 1;
    endif;
    EndDo;
    if Selected = 0 then
    Items.ТаблицяНоменклатури.AddRow();
    NewRow =
    Items.ТаблицяНоменклатури.CurrentData;

```

```

    NewRow.Номенклатура = SelectedValue;
    NewRow.Кількість = 1;
    НоменклатураOnChanged(SelectedValue);
    Items.ТаблицяНоменклатури.CurrentRow =
    Items.ТаблицяНоменклатури.CurrentRow + 1;
    else
    Message("Це найменування уже є в списку!");
    endif;
EndProcedure

```

```
//=====
```

2.10. Для форми документа допишіть обробку події *BeforeWrite*, у якій перевіряється заповнення реквізитів заголовку документа.

2.11. Для форми документа допишіть обробку події *OnOpen*, у якій за замовчуванням заповнюється назва організації, комірник та склад.

2.12. Створіть друковану форму документа.

3. Скопіюйте документ *НадходженняТоварівПослуг* і перейменуйте новий документ у *ПродажТоварівПослугПродукції*.

3.1. Зверніть увагу, що всі реквізити і табличні частини скопійувалися разом з документом, але у формі документа заголовки залишилися без змін. Перейменуйте його на *Продаж товарів, послуг, продукції*.

3.2. Оскільки при продажі використовують роздрібні чи гуртові ціни, то для поля *Номенклатура* допишіть обробку події *OnChange*, яка автоматично вставлятиме у табличну частину документа ціни вибраного найменування в залежності від **типу складу**. Для цього до функції додайте ще один параметр *Склад*

```
//=====
```

```

Function GetPrice(Склад, Номенклатура)
    priceType = Enums.ТипЦін.Закупівельна;

```

```

        if          Склад.ТипСкладу          =
Enums.ТипСкладу.роздрібний
then
priceType = Enums.ТипЦін.роздрібна
else
priceType = Enums.ТипЦін.гуртова;
endif;

        // далі без змін
EndFunction

//=====

```

Зауваження. У процедурі **НоменклатураOnChange(Item)** підпра-
вте виклик процедури **GetPrice()**.

4. Перейдіть на гілку *Журнали документів*.

5. Створіть журнал документів *РухНоменклатури*, у якому відображатимуться документи по надходженню та продажу товарів:

5.1. Створіть журнал документів *РухНоменклатури*.

5.2. На вкладці *Дані* додайте *Документи*, що реєструються журналом – два документи: *НадходженняТоварівПослуг* і *ПродажТоварівПослугПродукції*.

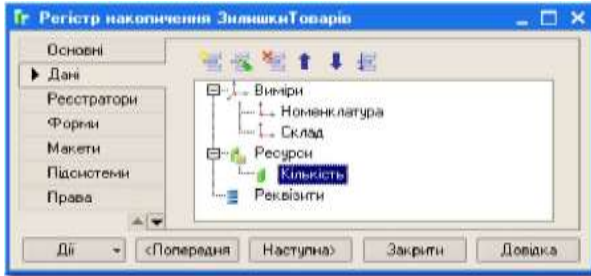
5.3. На вкладці *Дані* додайте графу *Контрагенти*. Після зазначення ім'я графи додайте *Посилання*, яке міститиме реквізит *Контрагенти* для кожного з документів.

5.4. Додайте у журнал документів *РухТоварів* графу *Сума* та зв'яжіть її посиланням з реквізитом *Сума* документа *НадходженняТоварівПослуг*;

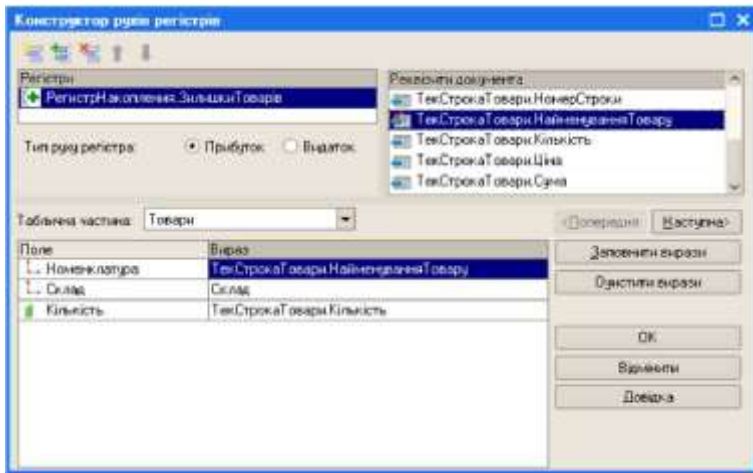
5.5. Виконайте те саме для документа *ПродажТоварівПослугПродукції*.

Завдання 2. Регістри накопичення. Проведення документів.

1. Відкрийте регістр накопичення *ЗалишкиТоварів* і перевірте, що задано тип *Залишки*.



2. Перейдіть у конфігурації до документа *НадходженняТоварів* і заповніть конструктор рухів за типом руху *Прибуток*.

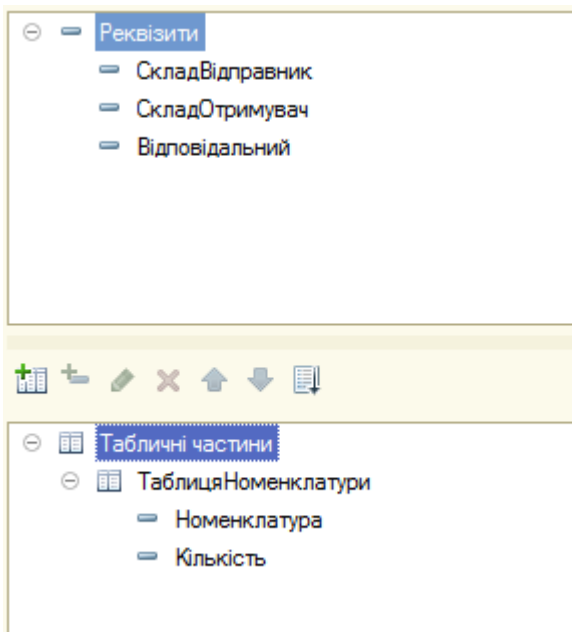


3. У документ додайте рух регістру накопичення *ЗалишкиТоварів* за типом руху *Видаток*.

4*. Напишіть спільну процедуру, в якій порівнюється зазначена кількість у формі з наявною на складі. Відкрийте модуль об'єкта *ПродажТоварів* на вкладці *Інше* конструктора документа. Додайте у процедуру *Posting()* перевірку кількості.

5. Скопіюйте документ *ПереміщенняНоменклатури*, за допомогою якого можна переміщувати товари з одного складу до іншого.

6. Вилучіть зайві реквізити та реквізити табличної частини. Аналогічно зайві процедури у формі.



7. Сформуйте рух реєстру накопичення *ЗалишкиНоменклатури* таким чином, щоб один документ одночасно формував видаток товарів з одного складу та надходження товарів на інший склад. Для цього:

- у документі *Переміщення* обрати реєстр *ЗалишкиНоменклатури*, рух у якому він змінюватиметься;
- викликати конструктор рухів (для виду рухів *Прибуток* (зверніть увагу на те, який склад слід вказувати при кожному виду руху));
- відкрити модуль об'єкта, що містить код процедури *Posting()* та скопіювати код проведення документа у буфер обміну;
- ще раз запустити конструктор рухів, але для виду руху *Видаток* (зверніть увагу на те, який склад слід вказувати при кожному виду руху);
- доповніть код обробника процедурою перевірки на кількість номенклатури на складі, звідки вона видається.

9. Створіть друковану форму документа. Пропишіть процедуру перевірки заповнення даних за необхідності.

10. Додайте документ *Переміщення* до журналу документів *Рух Номенклатури*.

11. Додайте у конфігурацію реєстр накопичення *ПродажіКомпанії* Встановіть для цього реєстру вид *обороти*. Сформуйте структуру реєстру таким чином, щоб він зберігав інформацію про те, якому контрагентові, коли і на яку суму був проданий товар.

12. Змініть процедуру проведення для документа *ПродажТоварівПослугПродукції*. Зверніть увагу, якщо запустити заново конструктор рухів, то всі зміни внесені в процедуру *Posting()* буде знищено. Отже, перед викликом конструктору рухів слід скопіювати у буфер обміну всю процедуру *Posting()*, після чого знову викликати конструктор і заново внести зміни в процедуру сформовану автоматично (використовуючи інформацію з буферу обміну).

Завдання 3. Заповнення інформаційної бази.

1. Збережіть конфігурацію.

2. Перейдіть у режим налагодження системи та заповніть інформаційну базу.

4. За допомогою *Синтаксис-Помічника* ознайомитись із синтаксисом і призначенням функцій, процедур, методів, які використовувались.

Завдання 4. Звітність

1. Вивантажте інформаційну базу у файл з розширенням *.dt*, який перейменуйте й у назву до прізвища додайте номер лабораторної роботи.
2. Завантажте файл та звіт з переліком виконавців та виконаних видів робіт.

Запитання для повторення

1. Які види реєстрів накопичення є?
Яка між ними відмінність?
2. Які є види кнопок можна створювати у фреймворку?
Як організувати події для них?

3. Призначення журналів документів. Яка їхня відмінність від списків документів?
4. Чи може документ належати до декількох журналів?
5. Дані яких реквізитів документів можна виводити у графах журналів документів?

Лабораторна робота № 5

Створення конфігурації: Звіти. Обробки

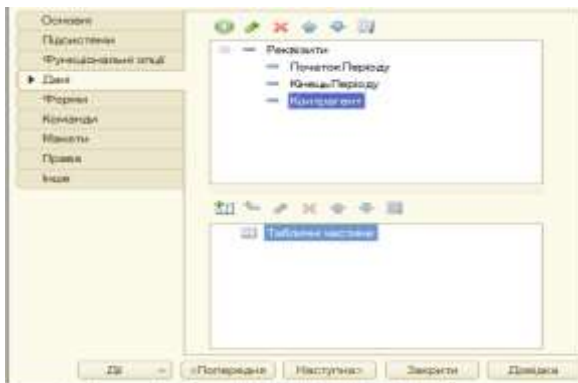
Завдання:

1. Створити звіти та обробки, зазначені у таблиці та відповідно до свого варіанту.
2. Прописати модулі форм та об'єктів.
3. Відредагувати інтерфейси користувачів.
4. Перевірити роботу створених об'єктів, перейшовши у режим налагодження роботи
5. Заповнити інформаційну базу.
6. Вивантажити файл .dt
7. Захистити роботу.
8. Завантажити звіт та архів на сторінку курсу.

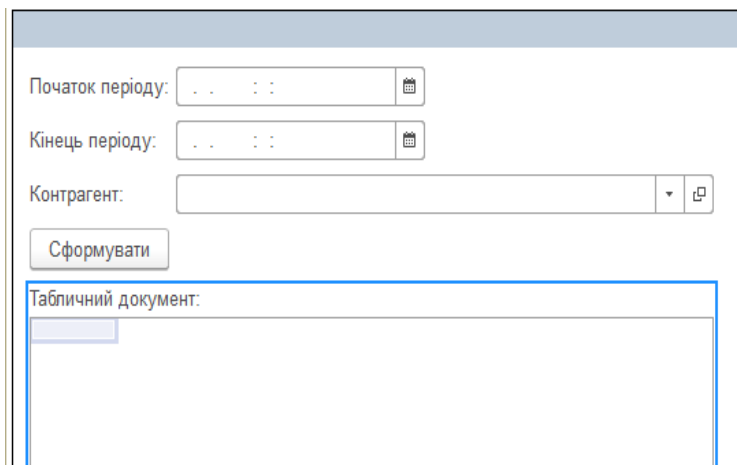
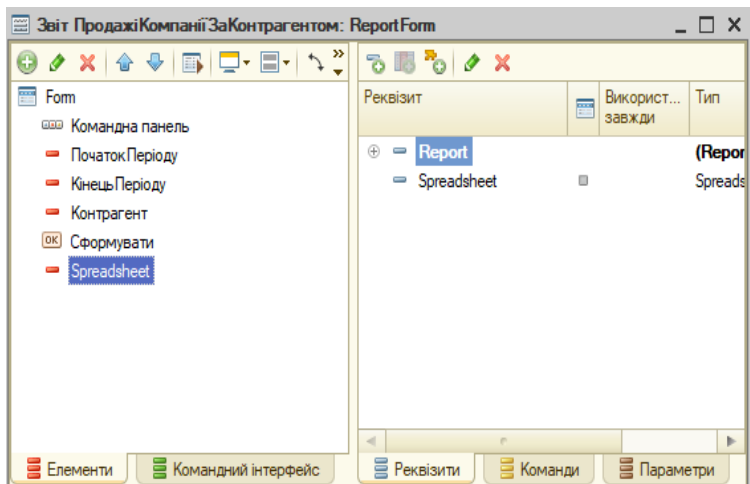
Хід роботи:

Завдання 1. Звіти.

1. Створюємо звіт *Продажі компанії за контрагентом*
 - 1.1. Додаємо дані



1.2. Створюємо форму звіту, на яку додаємо табличний документ



1.3. Для команди *Сформувати* прописуємо обробник.

Зауваження. 1. При побудові коду використовувати конструктор запиту з обробкою результату. Доповнити необхідними конструкціями.

```

//=====
&AtServer

Function СформуватиAtServer(Контрагент, Поча-
токПеріоду, КінецьПеріоду)
// Вставити вміст обробника.
//{{QUERY_BUILDER_WITH_RESULT_PROCESSING
// Даний фрагмент побудований конструктором.
// При повторному використанні конструктора,
внесені вручну зміни будуть втрачені!!!

Template =
Reports.ПродажіКомпаніїЗаКонтрагентом.GetTe-
mplate("ПродажіКонтрагенту");

Query = New Query;

Query.Text =
    "SELECT
        |      ПродажіКомпані.Контрагент AS Ко-
нтрагент,
        |      ПродажіКомпані.Сума AS Сума,
        |      ПродажіКомпані.Period AS Period
    |FROM
        |
        AccumulationRegister.ПродажіКомпані AS
ПродажіКомпані
        |WHERE
        |      ПродажіКомпані.Контрагент =
&Контрагент And

```

```

        |      ПродажіКомпані.Period >=
&ПочатокПеріоду
        |      AND ПродажіКомпані.Period <=
&КінецьПеріоду";

Query.SetParameter("Контрагент", Контра-
гент);

Query.SetParameter("КінецьПеріоду", Кінець-
Періоду);

Query.SetParameter("ПочатокПеріоду", Поча-
токПеріоду);

QueryResult = Query.Execute();

HeaderArea = Template.GetArea("Header");
FooterArea = Template.GetArea("Footer");
TableHeaderArea =
Template.GetArea("TableHeader");
TableFooterArea =
Template.GetArea("TableFooter");
DetailRecordArea =
Template.GetArea("Details");

Spreadsheet.Clear();
Spreadsheet.Put(HeaderArea);
Spreadsheet.Put(TableHeaderArea);
Spreadsheet.StartRowAutoGrouping();
SelectionDetailRecords =
QueryResult.Select();

```



```

While SelectionDetailRecords.Next() Do

    DetailRecordArea.Parameters.Fill(SelectionDetailRecords);

    Spreadsheet.Put(DetailRecordArea,
SelectionDetailRecords.Level());

EndDo;

Spreadsheet.EndRowAutoGrouping();
Spreadsheet.Put(TableFooterArea);
Spreadsheet.Put(FooterArea);

//}}QUERY_BUILDER_WITH_RESULT_PROCESSING
return Spreadsheet;
EndFunction

&AtClient
Procedure Сформувати(Command)

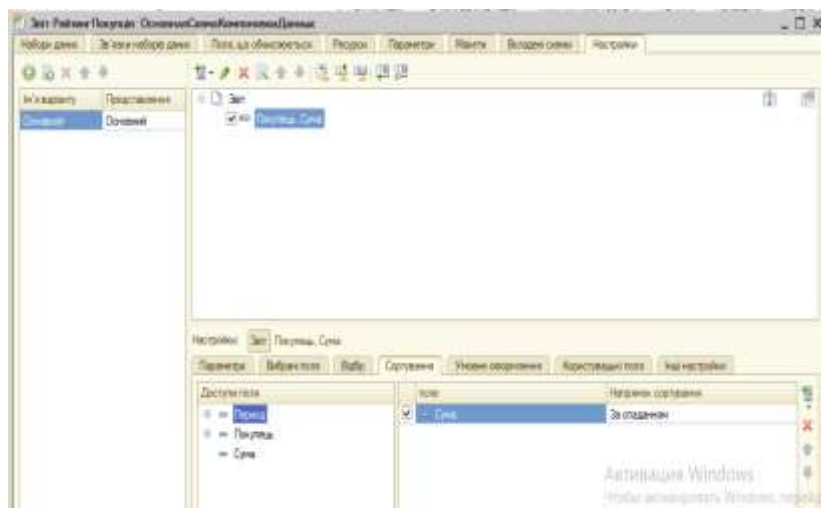
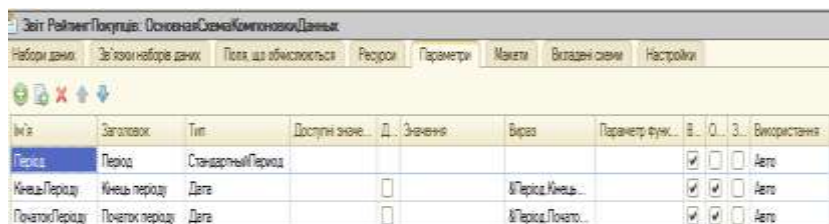
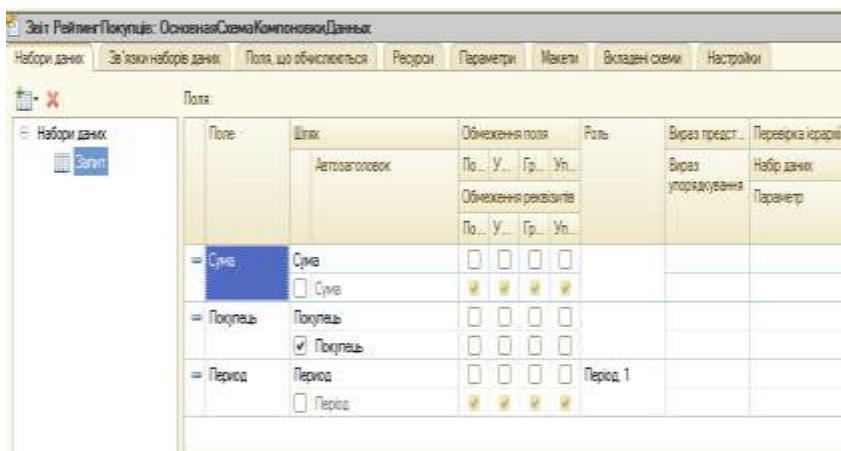
    Spreadsheet = Сформува-
тиAtServer(Report.Контрагент, Report.Початок
Періоду, Report.КінецьПеріоду);

EndProcedure

//=====

```

2. Створюємо звіт *Рейтинг покупців* за допомогою схеми *компоновки даних*. При цьому користуємось конструктором запиту в середині конструктора.



3. Аналогічно створюємо звіт *Найпопулярніша номенклатура* за допомогою *схеми компоновки даних*

4. Створюємо звіт *Залишки номенклатури на складі*, який використовує дані з реєстру накопичення *.ЗалишкиНоменклатури* виду залишки. При цьому *Номенклатура*, *Склад*, *Початок періоду*, *Кінець періоду* є параметрами.

5*. Передбачити можливість виведення інформації у розрізі задання/ не задання значень параметрів.

Завдання 2. Обробки (Самостійне опрацювання).

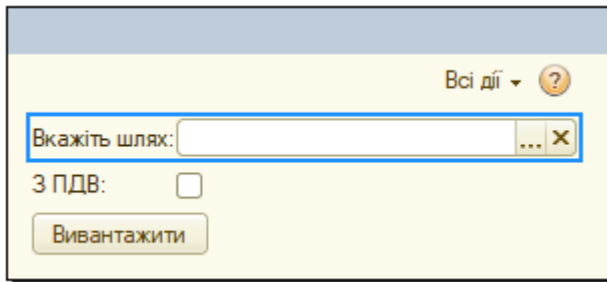
1. Створюємо обробку *Прайс лист*, яка вивантажує ціни номенклатури

1.1 Перейти на гілку *Обробки* та задати ім'я створеній обробці.

1.2 На вкладці *даних* задати реквізит *Ім'яШляху* (тип: *Рядок*)

1.3 Створити форму обробки.

1.4 Створити кнопку вивантажити аналогічно як кнопку *Сформувати* для звітів



1.5. Створюємо код події кнопки *Вивантажити* за допомогою конструктора побудови запитів з обробкою результату.

2. Доопрацьовуємо шаблон макету, створений конструктором.

| № | Номенклатура | Роздрібна ціна, грн | Гуртова ціна, грн |
|---------|----------------|---------------------|-------------------|
| <номер> | <номенклатура> | <роздрібна> | <гуртова> |

- Опрацювати за допомогою синтакс-помічника роботу з об'єктами:

RunApp:

FileDialogMode;
FileDialog

і напишіть код на стороні клієнта вивантаження даних у документ.

Зауваження. Організацію можна оформити як контрагента.

Завдання 4. Звітність

- Вивантажте інформаційну базу у файл з розширенням .dt, який перейменуйте й у назву до прізвища додайте номер лабораторної роботи.
- Завантажте файл та звіт з переліком виконавців та виконаних видів робіт.

Запитання для повторення

- Яке призначення звітів?
- Якими способами можна створювати звіти?
- Куди можна виводити результат звіту?
- Як налаштувати звіт схемою компоновки даних?
- Призначення обробок? Наведіть приклади.

Література

1. Шквір В. Д. Інформаційні системи і технології в обліку та аудиті: підручник / В. Д. Шквір, А. Г. Загородній, О. С. Височан. – 3-тє видання, стереотипне. – Львів : Вид-во Львівської політехніки, 2016. – 400 с.
2. Освітня програма «Інформаційні технології та управління проектами першого (бакалаврського) рівня вищої освіти» [Електронний ресурс] – Режим доступу : https://mathmod.chnu.edu.ua/media/1721/op_knbak_2021.pdf
3. Освітня програма «Системний аналіз першого (бакалаврського) рівня вищої освіти» [Електронний ресурс] – Режим доступу : https://mathmod.chnu.edu.ua/media/1723/op_sabak_2021.pdf
4. Демоверсії BAS. [Електронний ресурс] – Режим доступу : <https://www.netsoft.com.ua/demo.html>

Навчальне видання

Готинчан Тетяна Іванівна

Інформаційні системи обліку та логістика

**Методичні вказівки та завдання
до лабораторних робіт**

Відповідальний за випуск **Черевко І.М.**
Комп'ютерний набір **Готинчан Т.І.**