

Міністерство агропромислового комплексу України
Вінницький державний сільськогосподарський інститут

Кафедра інформатики та
автоматизації процесів
управління

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторного практикуму з дисципліни
"Програмне забезпечення ЕОМ"

Частина 2: БАЗИ ДАНИХ ТА СУБД (FoxPro 2.0)
для студентів спеціальностей
70.50.202, 70.50.106

Вінниця. 1997.

Методичні вказівки до виконання лабораторного практикуму з дисципліни "Програмне забезпечення ЕОМ", частина 2: Бази даних та СУБД (FoxPro 2.0) для студентів спеціальностей 70.50.202, 70.50.106. /Укладачі О.В.Бісікало, В.А.Паламарчук, С.Ф.Роботко, О.В.Бісікало. - Вінниця, ВДСГІ, 1997 р.. 64 с./.

У другій частині лабораторного практикуму з дисципліни "Програмне забезпечення ЕОМ" наведено методичні вказівки до виконання вісьми лабораторних робіт, спрямованих на вивчення баз даних та СУБД на прикладі популярної системи FoxPro 2.0. Розраховані на 44 години завдання лабораторного практикуму пов'язані за допомогою наскрізних об'єктів агропромислового спрямування, а також проілюстровані наскрізним прикладом додатку до бази даних з умовною назвою "колгоспна ідальня". В результаті послідовного виконання всіх лабораторних робіт студенти отримують повне програмне забезпечення додатку до власної бази даних.

Навчальне видання переглянуто та ухвалено на засіданні кафедри інформатики та АІУ ВДСГІ. Протокол №10 від 1 липня 1997 р.

Друкуються на основі рішення Вченої Ради ВДСГІ. Протокол №3 від 30 жовтня 1997 р.

- 1 -

Загальні вимоги до виконання та оформлення
лабораторних робіт

Лабораторний практикум з дисципліни "Програмне забезпечення (частина 2): БАЗИ ДАНИХ ТА СУБД (FoxPro 2.0)" відповідає півній наскрізної комп'ютерної підготовки, виконаної авторським колективом кафедри інформатики та АБУ ВАСГІ. Методичні вказівки до виконання лабораторних робіт складено у вигляді чітких слідovenних покрокових інструкцій, проілюстрованих наскрізним прикладом умовного об'єкту автоматизації - колгоспної ідеальної студент обирає індивідуальний варіант об'єкту агропального спрямування (с.) на підставі власного номеру в літ групи. Цей варіант об'єкту автоматизації має бути для нта наскрізним для всіх робіт лабораторного практикуму. Токонання кожної лабораторної роботи необхідно проводити за лном стандартною методикою:

- до початку пари треба ознайомитися з методичними вказівками до виконання лабораторної роботи та зробити заготовку відного звіту: заготовка обов'язково повинна починатися з тіння:

ва: "Звіт по лабораторній роботі № ..."

на лабораторної роботи:

на роботи:

- необхідності: звіт можна також записати деякі потрібні дані з списку теоретичних відомостей:

- оскільки всі завдання лабораторного практикуму пов'язані зовсім одним об'єктом, а кінцевим результатом загального циклу звіт має бути власний додаток до бази даних, всі без винятку к необхідно послідовно оформлювати в окремому звітні - аку- з. з отриманням вимог відповідних стандартів (ЕСХД, ЕСБД):

- до початку пари варто уважно продивитися зміст роботи та нати її дії (по аналогії з наскрізним прикладом), які не вбукте використання комп'ютера - намалювати ескіз, розробити проєкти структурних блок-схеми алгоритмів, написати тексти івних програмних процедур, тощо;

- найголовнішим результатом самостійної підготовки до лабораторної роботи має бути чітке розуміння того, що і як треба робити парі: - тільки це уявлення студента разом з якимсь експертним заготовкою врахатися єдиною основою допуску його виклада-

чем до лабораторної роботи:

- одержавши на початку пари допуск до лабораторної роботи, необхідно послідовно виконати всі пункти змісту роботи, відображавши отримані результати та зроблені висновки у звіті: якщо кінцеві (відлагоджені) програми чи інші результати відрізняються від попередніх, слід звернути на це увагу в звіті;

- після виконання змісту роботи необхідно переглянути розділ методичних вказівок "Зміст звіту" та пересвідчитися в тому, Ваш звіт повністю відповідає останньому;

- звіт по лабораторній роботі обов'язково треба закінчити загальними висновками, в яких слід підсумувати всі результати, з'ясувати повноту досягнення мети роботи, оцінити користь отриманих знань та набутих навичок з точки зору майбутньої спеціальності, тощо;

- завершується процес виконання лабораторної роботи її захистом у викладача: до захисту треба мати повний звіт, розумітися на всіх отриманих у ході виконання роботи результатах і вміти їх продемонструвати на комп'ютері, а також знати лекційний матеріал по темі роботи та зміти відповідати на всі питання самоконтролю.

ЛАБОРАТОРНА РОБОТА № 2.1

ТЕМА: ВИВЧЕННЯ ОСОБЛИВОСТЕЙ ІНТЕРФЕЙСУ ІНТЕРПРЕТАТОРА
FoxPro 2.0 (4 год.).

МЕТА РОБОТИ: ознайомлення з головним меню системи FoxPro та основними поняттями багатовіконного інтерфейсу; набуття навичок роботи з загальними можливостями інтерпретатора системи FoxPro, що не стосуються баз даних.

ЗМІСТ РОБОТИ

1. При необхідності підключити мишу до послідовного порту комп'ютера та запустити драйвер миші (для мережі Z:\WINDOWS\

LMOUSE.COM); перейти на диск c:\ та через головне меню Norton Commander зайти в систему FoxPro.

2. В головному меню системи FoxPro знайти і послідовно запустити наступні опції File ! Open, вказати тип (Type) File, вибрати диск C та відкрити будь-який текстовий файл на ньому. Якщо на Вашому мережевому диску C:\ немає придатного для експериментування файлу, необхідно зайти у режим System ! Filer, знайти будь-який текстовий файл у розділі Z:\METHOD\ і скопіювати його на свій диск (файл позначається клавішею Space, потім вибирається режим Copy, в якому для Target directory треба вибрати директорию-приймач файлу).

3. Перевірити у режимі System ! Filer як виконуються основні операції з файлами - копіювання, перенос, перейменування, знищення, сортування, пошук, зміна атрибутів файлу та малювання дерева (позначення групи файлів - комбінація Shift-Space). Порівняти можливості файлеру FoxPro з можливостями оболонки Norton Commander. Дати опис своїх дій та викласти зроблені висновки у звіті.

4. На створеному вікні з відкритим текстовим файлом перевірити всі можливі операції з меню, такі як :

- горизонтальне пересування тексту у вікні;
- вертикальне пересування тексту у вікні;
- пересування вікна по екрану монітора;
- зміна розміру вікна;
- перехід до повного екрану і зворотня операція;
- закриття вікна.

Виконати всі ті самі операції з вікном за допомогою опції Window головного меню системи та занотувати у звіт відповідні "гарячі комбінації" клавіш.

5. Зайти у опцію головного меню System, де активізувати ще два вікна : Calculator та Puzzle. За допомогою меню розташувати всі чотири вікна (в тому числі командне вікно Command) на екрані на свій розсуд. Спробувати також зробити переходи між вікнами простим натисканням клавіші миші. Порівняти можливі дії п.3 з усіма розташованими на екрані вікнами. Описати свої дії у звіті.

6. Перейти у вікно з відкритим текстовим файлом і перейменувати його під назвою ZVIT1.TXT за допомогою опції File ! Save as... . Правильність Ваших дій покаже нова назва активного вікна. Тепер метом наступних Ваших дій має бути створення влас-

ного звіту з попереднього тексту у відповідності з розділом ВМІСТ ЗВІТУ. Це забезпечить вивчення наступних стандартних операцій вбудованого редактора FoxPro:

- позначення блока тексту (Shift + стрілочки) або довгий клік миші (натискання лівої кнопки на початку блока, пересування миші на кінець блока, де кнопку можна відпустити);
- запис позначеного тексту в "кишені" оперативної пам'яті за допомогою Edit ! Cut або Edit ! Copy;
- виймання позначеного тексту з "кишені" оперативної пам'яті за допомогою Edit ! Paste;
- знищення позначеного в блок тексту за допомогою клавіші Del.

Занотувати в звіт "гарячі комбінації" клавіш, що відповідають приведеним вище діям.

7. Визвати клавішею F1 допомогу у різних режимах головного меню FoxPro. Що Ви побачили ? Спробуйте в тексті Help'у, що з'явився записати частину інформації в блок та перенести його в свій звіт. Висновки щодо контекстності та гіпертекстності Help'у також занесіть в звіт.

8. Перейти у вікно "Command" за допомогою комбінації Ctrl+F2 і подивитися, що там є. Стрілочкою стати на попередню команду Help та натиснути Enter. Спробувати також викликати інші команди, що є у вікні "Command". Зробити висновки щодо відповідності виклику команди через головне меню інтерпретатору та вікно Command.

9. У вікні "Command" набрати команду !ps . Що Ви бачите? Спробувати викликати інші DOS'івські команди через окличний знак у вікні "Command", наприклад, !LEX, !ST, !Q тощо. Якщо команда не виконується, необхідно вказати явно шлях знаходження відповідного *.exe або *.com файлу. Висновки занотуйте в звіт.

10. За допомогою HELP'у визначити дії опцій File ! Printer Setup та File ! Print . Занотувати у звіт можливості системи FoxPro щодо підготовки документів до друку.

11. За допомогою HELP'у визначити дії опцій Window ! Color. Занотувати у звіт можливості системи FoxPro щодо зміни кольорів основних елементів інтерфейсу.

12. За допомогою HELP'у визначити дії опцій System ! Macros. Занотувати у звіт можливості системи FoxPro щодо створення такого виду електронної презентації, як демонстраційна версія програмного засобу чи послідовності дій в системі.

13. Вийти з системи Foxpro за допомогою команди Quit у вікні "Command" або відповідної опції головного меню (File : Quit).

ВМІСТ ЗВІТУ

Звіт з лабораторної роботи обов'язково повинен вмішувати:
опис можливих операцій з файлами за допомогою режиму System : Filer згідно п. 3 змісту роботи;
опис можливих операцій з вікнами за допомогою "миші" згідно пп. 4 та 5 змісту роботи;
відмінності між можливими діями з різними типами вікон;
"гарячі" комбінації клавіш, що відповідають режимам Cut, Copy та Paste;
фрагмент будь-якого тексту з Help'у та висновки щодо нього згідно п.7 змісту роботи;
висновки щодо паралельності та тотожності виклику команд в інтерпретаторі через головне меню та вікно "Command";
висновки щодо виклику DOS'івських команд через вікно "Command";
опис можливостей системи згідно з пп. 10, 11, 12.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Як можна використати маніпулятор миша в системі FoxPro?
2. Які основні режими роботи (опції головного меню) має система FoxPro?
3. Які операції над файлами можна виконати в режимі System : Filer?
4. Які операції можна робити з вікнами за допомогою "миші"?
5. Наведіть відмінності між можливими діями з різними типами вікон в системі Foxpro.
6. Які "гарячі" комбінації клавіш відповідають режимам Cut, Copy та Paste?
7. Які принципи побудови Help'у в системі Foxpro Вам відомі?
8. Як забезпечується паралельність та тотожність виклику команд інтерпретатора через головне меню та через вікно "Command"?
9. Які Ви вивчили прийоми позначення блоку інформації, його копіювання та переносу в системі FoxPro?

10. В чому полягає особливість виклику DOS-команд на відміну від команд FoxPro через вікно "Command"?
11. Які загальні можливості багатовіконного інтерфейсу підтримує система FoxPro?
12. Що таке макрос? Для чого використовують макроси? Як створити макрос в системі FoxPro?
13. Які параметри друку дозволяє регулювати FoxPro?

ЛАБОРАТОРНА РОБОТА №2.2

ТЕМА: СТВОРЕННЯ ВЗАЄМОПОВ'ЯЗАНІ СТРУКТУРИ БАЗИ ДАНИХ У ВИГЛЯДІ *.DBF-ФАЙЛІВ ТА ЗАПОВНЕННЯ ЇЇ ДАНИМИ (6 год.).

МЕТА РОБОТИ: ознайомлення з методикою створення *.dbf-файлів в системі FoxPro та основними поняттями бази даних; набуття навичок роботи проектування та заповнення баз даних в режимі Browse.

СТИСЛІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Створення та підготовка будь-яких баз даних для подальшої обробки вимагає ретельного планування їх структури (записів, полів, тобто всіх атрибутів). Характерно, що при цьому майже завжди проявляється ефект збитковості, коли в базі багато разів повторюються одні і ті самі дані (наприклад, в базі даних телефонного довідника це - назви вулиць, назви установ, назви підрозділів установи). Збитковість бази, по-перше, суттєво збільшує її обсяг, по-друге, не менш суттєво ускладнює її редагування, а, по-третє, вагомо збільшується час пошуку потрібних даних. Як зробити так, щоб база даних була організована раціонально?

Головними критеріями оптимальності при побудові бази даних вважаються мінімум дискової пам'яті, що займає база, та мінімум середнього часу доступу до її даних. Отже, база даних тим краща, чим менший обсяг пам'яті вона займає і чим швидше дозволяє знайти потрібні дані. Для вирішення окресленої проблеми було розроблено математичний апарат нормальних форм Кодда (інша назва - нормалізація відношень), який з успіхом використовується в

сучасних реляційних базах даних та системах управління базами даних (СУБД). Суть цього методу оптимізації баз даних полягає в послідовному розкладенні вихідних відношень (таблиць) бази на інші, більш мілкі та прості відношення. При цьому замість назв або інших даних, що повторюються, ставиться лише номер запису відповідного довідника, який знаходиться в іншій таблиці (*.dbf-файлі). Посилання на іншу таблицю можна також забезпечити за допомогою спеціальних шифрів - це допомагає уникнути складнощів в ситуаціях, коли довідник часто модифікується (зникають, додаються чи змінюють записи). Для випадку бази даних з номерами телефонів замість назв вулиць (40 знаків), наприклад, можна використати лише їх порядкові номери (4 знаки) з додаткового довідника вулиць міста (WorldBestTown, тобто Вінниці).

Процес ділення однієї таблиці на декілька називається декомпозицією (розкладенням). Необхідно підкреслити, що даний процес є покроковим та оборотним, а кожний наступний його крок виявляє та знешкоджує небажані функційні залежності між атрибутами бази даних. Теорія нормалізації відношень, яка була розроблена Е.Ф.Коддом, досить складна, а тому використовується лише професійними програмістами в галузі баз даних. Для практики створення додатків до баз даних економічного призначення досить знати загальні принципи побудови відношень у так званій формі Бойса-Кодда. Іншими словами, непростий процес розкладення початкових відношень у потрібні нормальні форми замінюється методикою побудови бази даних, що задовільняє вимогам оптимальності. Стисло практичну методику створення бази даних можна викласти у вигляді наступних кроків:

- слід визначити всі групи даних та існуючі відношення між ними, які необхідні для вирішення задачі (інакше - всі інформації для кожного з документів, що використовується у задачі);
- спробувати представити всі дані задачі у вигляді єдиної таблиці (головного файлу) або, якщо це неможливо, кількох великих таблиць;
- ті групи (поля) даних, що повторюються у головному файлі, необхідно замінити одним полем з шифром, а самі дані винести в спеціальний файл-довідник (в довіднику має бути таке ж саме поле з шифром для зв'язку між ним та головним файлом);
- операції декомпозиції послідовно проводити з головним файлом та файлами-довідниками до тих пір, поки в кожному з отри-

маних *.dbf-файлів не залишаться тільки унікальні дані, тобто такі дані, які принципово, по своїй природі не можуть дублюватися в різних записах (наприклад - прізвище, ім'я та по-батькові; номер та дата бухгалтерського документу; банківські атрибути).

В результаті виконання окресленої методики можна отримати базу даних у вигляді так званого "розсіпу *.dbf-файлів", пов'язаних між собою за допомогою додаткових полів з шифрами. Цим самим досягається значна економія дискової пам'яті, тому що у файлах з найбільшою кількістю записів дублюються лише порівняно невеликі поля (з шифрами), а основна довідкова інформація зберігається окремо. При цьому зменшується також і середній час доступу до таких даних, оскільки їх обсяг на диску суттєво менший, і, відповідно, апаратні засоби ЛЕОМ швидше їх можуть обробити.

З М І С Т Р О Б О Т И

1. Створити проект бази даних по своєму варіанту у вигляді взаємопов'язаної структури *.DBF-файлів. Для цього скористуватись індивідуальним завданням у вигляді об'єкту АПК, що надалі буде використовуватись як наскрізний об'єкт, з таблиці п.9. Методичних вказівок.

При розробці бази даних необхідно так, як і в наведеному нижче прикладі, запропонувати структуру головного файлу та 3-х або більше файлів-довідників. Проект бази даних треба обов'язково занотувати у звіт та узгодити з викладачем, якому необхідно довести, виходячи з специфіки свого об'єкту, доцільність типу та розміру кожного з запропонованих полів.

1.1. Виконання лабораторної роботи розглянемо на прикладі кооперативної ідальні, де ведеться облік продуктів і тари. Вхідними документами вважаються накладні, а вихідним документом є реєстр товарів. Пропонується створити базу даних з 5-ти взаємопов'язаних *.dbf-файлів, куди входять:

а) Головний файл (RE_PROD.DBF) - реєстр товарів, що має таку структуру:

- 10 -

Field	Field Name	Type	Width	Dec	Comment
(Name)	(Im'a nazv)	(Typ)	(Poslup)	(Dec.)	(Primitka)
1	NA_PROD	Character	30		назва продукту
2	SH_NAKL	Numeric	4		шифр накладної
3	SH_OD_VYM	Numeric	2		шифр одиниці виміру
4	KILKIST	Numeric	10	3	кількість товару
5	CINA	Numeric	10	2	ціна товару
** Total **			57		

Примітка: шифри використовуються для виокремлення такої ситуації, коли в багатьох записах повторюється одна і та сама інформація (наприклад, номер і дата накладної, одиниця виміру тощо). Кожному шифру відповідає спеціальний файл-довідник, причому сумарний обсяг інформації, яка відводиться під нього та поле з шифром, набагато менший від того обсягу інформації, який би знадобився для неодноразового повторення в головному файлі однієї інформації;

б) Довідник одиниць виміру (D_OD_VYM.DBF) - має наступну структуру:

Field	Field Name	Type	Width	Dec	Comment
1	SHIFR_OD	Numeric	2		шифр одиниці виміру
2	NA_OD_VYM	Character	20		назва один. виміру
3	SH_BAZ_OD	Numeric	2		шифр базової одиниці
4	KOEFFIC	Numeric	8	3	коефіц. перерахунку
** Total **			33		

Примітка: шифри в головному файлі та в файлі-довіднику обов'язково використовуються такі ж самі з метою створення відповідних зв'язків між цими файлами;

в) Довідник накладних (D_NAKL.DBF) - має наступну структуру:

Field	Field Name	Type	Width	Dec	Comment
1	SH_NAKL	Numeric	4		шифр накладної
2	DATA_NAKL	Date	8		дата
3	NOM_NAKL	Numeric	4		номер
4	SH_POSTACH	Numeric	3		шифр постачальника
** Total **			19		

г) Довідник постачальників (D_POSTAC.DBF) - має наступну структуру:

- 11 -

Field	Field Name	Type	Width	Dec	Comment
1	SH_POSTACH	Numeric	3		шифр постачальника
2	NA_POST	Character	30		назва постачальника
3	ROZR_RAH	Numeric	10		розрахунок, рахунок
4	SH_BANKA	Numeric	2		шифр банку
5	UR_ADRESA	Memo	10		юридична адреса
6	TELEF	Numeric	7		телефон
7	N_PODATK	Numeric	8		податковий номер
** Total **			71		

д) Довідник банків (D_BANKA.DBF) - має таку структуру:

Field	Field Name	Type	Width	Dec	Comment
1	SH_BANKA	Numeric	2		шифр банку
2	NA_BANKA	Character	30		назва банку
3	MFO	Numeric	6		МФО
** Total **			39		

1.2. Придумати вірогідні дані контрольного прикладу для всіх файлів, що входять до бази даних. Занотувати ці дані в звіт у вигляді відповідних таблиць.

Приклад колгоспної їдальні. Припустимо, що на звітний період маємо такі дані по файлам п.1.2.:

а) Головний файл - реєстр товарів - має наступний вигляд:

Record#	NA_PROD	SH_NAKL	SH_OD_VYM	KILKIST	CINA
1	Картопля	1	5	2.100	300.00
2	Буряк	1	1	520.000	0.25
3	М'ясо свин.	2	1	125.000	5.20
4	М'ясо волоче	2	1	68.000	3.40
5	Морква	1	1	54.000	0.15
6	Ковбаса московська	2	1	42.000	9.30
7	Сметана	3	3	40.000	1.80
8	Молоко	3	3	40.000	0.55
9	Вершки	3	3	40.000	1.20
10	Хліб чорний	4	1	74.000	0.68
11	Сіль	5	4	1.200	250.00
12	Яйця курячі	6	2	980.000	0.16
13	Контейнери	1	2	12.000	7.30
14	Яшки	2	2	22.000	3.50
15	Бідони	3	2	3.000	8.80

б) Довідник одиниць виміру має наступний вигляд:

Record#	SHIFR_OD	NA_OD_VYM	SH_BAZ_OD	KOEFFIC
1	1	кілограм		0.000
2	2	штука		0.000
3	3	літр		0.000
4	4	грам	1	1000.000
5	5	тона	1	0.001

в) Довідник накладних має наступний вигляд:

Record#	SH_NAKL	DATA_NAKL	NOM_NAKL	SH_POSTACH
1	1	03.02.97	23	2
2	2	03.03.97	15	1
3	3	03.03.97	4	4
4	4	28.02.97	87	3
5	5	02.03.97	11	5

г) Довідник постачальників має наступний вигляд:

Record#	SH_POST	NA_POST	ROZR_RAH	SH_BA	UR_AD	TELEF
1	1	Районний м'ясокомбінат	124512	1	немо	221820
2	2	КСЯ "Південнобузьке"	2345642	2	немо	22412
3	3	Хлібзавод	21654654	3	немо	51206
4	4	Молокозавод	1123456	1	немо	55612
5	5	Райспоживспілка	3235645	3	немо	12124

д) Довідник банків має такий вигляд:

Record#	SH_BANKA	NA_BANKA	MFO
1	1	Промінвестбанк м.Вінниці	302520
2	2	Обл.управління нацбанку України	302010
3	3	Томашп.відділ агропромбанку	305212

1.3. Після створення подібного проекту на папері занесіть всі дані в файли. Для цього знайдіть у головному меню File/New, помітьте клавішою Space тип Database і натисніть на кнопку <OK>. Процес введення даних щодо структури головного *.dbf-файлу бази згідно з п.1.1. інтуїтивно зрозумілий. При відсутності мими пересування між полями для введення інформації забезпечується клавішею Tab. Після занесення імені файлу введіть в нього дані з п.1.2. В процесі створення структури викликайте допомогу F1 і занотуйте в звіт послідовність Ваших дій та реакцій на них системи при введенні відповідних даних.

1.4. Повторити дії п.1.3. (окрім викликання допомоги) для всіх файлів-довідників, що включено Вами до проекту.

2. У командному вікні (Ctrl+F2) дати команду Use. Тепер через головне меню (File : Open) відкрити головний файл бази, а

потім передивитися його через режим Database : Browse. Стати на останній запис бази даних і, через опції головного меню Record : Append, ввести новий запис. Зверніть увагу на те, як ці команди продублювалися у командному вікні. Висновки занотуйте у звіт.

3. Занотуйте у звіт результати роботи інших режимів Record, а також всіх режимів опції Browse, що виникла в головному меню, користуючись, при необхідності, допомогою F1. Перевірте, як виконуються всі ці режими для файлів-довідників бази даних. Чи є відмінності? Висновки занесіть в звіт.

4. Відкрийте командою Use (Im'я *.dbf-файлу) в командному вікні (або через головне меню: File : Open) головний файл бази. Створіть для нього файл a1.txt з даними про структуру за допомогою команди: list stru to a1.

5. Відкрийте файл a1.txt у іншому вікні. Відмітьте його блоком, а потім перенесіть у відкритий перед цим звітний файл zvit2.txt.

6. Дайте команду: list to a1, за допомогою якої у файл a1.txt занесуться дані записів (інформація) відкритого на цей момент *.dbf-файлу. Перейдіть у відкритий в одному з вікон файл a1.txt, відмітьте його як блок, а потім допишіть у відкритий в іншому вікні звітний файл zvit2.txt.

7. Повторіть дії пп.4.-6. для всіх файлів-довідників бази. Зробіть порівняння первинної інформації в звіті про структуру та записи всіх файлів бази з даними файлу zvit10.txt. Висновки занесіть у звіт.

В М І С Т З В І Т У

Звіт з лабораторної роботи обов'язково повинен вмішувати:

- опис вибраної задачі Вашого об'єкту (бухгалтерської, економічної або юридичної) з вхідними та вихідними документами;
- проект бази даних по визначеній задачі;
- дані виконаного контрольного прикладу для всіх файлів бази;
- методика створення структури *.dbf-файлу у вигляді послідовності Ваших дій та відгуків на них системи;
- методика внесення даних в режимі Browse з позначенням можливих дій користувача;
- файл zvit2.txt у електронній формі або на твердому носіїві.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Що таке база даних? Як Ви розумієте поняття dBase-машини?
2. Яка особливість методики створення бази даних на FoxPro?
3. Які основні типи даних використовуються у СУБД FoxPro? Перерахуйте основні типи файлів.
4. Як створити структуру *.dbf-файлу?
5. Як заносити інформацію у базу даних в режимі Browse?
6. Які основні можливості режиму Browse Ви знаєте?
7. Що таке запис та чим він відрізняється від поля?
8. Які можливості надає режим Record?
9. Які команди треба виконати, щоб перенести в текстовий файл структуру та записи бази даних?
10. Для чого використовуються шифри та файли-довідники?

ЛАБОРАТОРНА РОБОТА №2.3

ТЕМА: СТВОРЕННЯ РЕЛЯЦІЙНИХ ЗВ'ЯЗКІВ МІЖ *.DBF-ФАЙЛАМИ БАЗИ ДАНИХ ТА ДОСЛІДЖЕННЯ ОСНОВНИХ КОМАНД НАВІГАЦІЇ (6 год.).

МЕТА РОБОТИ: ознайомлення з методикою створення реляційних зв'язків між *.dbf-файлами в системі FoxPro та набуття навичок навігації по базі даних в багатовіковому середовищі.

СТИСЛІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Команди навігації відносяться до найголовніших можливостей сучасних реляційних СУБД. За допомогою цих команд можливо пересуватися по записах бази даних таким чином: перейти на перший запис (початок бази) або останній запис (кінець бази), зміститися на наступний або попередній запис. В загальному випадку потрапити на будь-який запис бази можна двома шляхами - абсолютним (коли відомий номер запису) чи відносним (коли відоме положення потрібного запису відносно поточного запису).

Реляційні СУБД, в тому числі і FoxPro, мають стандартні можливості по заміні інформації в базі даних. До них належать

додавання нового запису в кінець бази даних або в її середину (після поточного запису), логічне або фізичне знищення потрібного запису та заповнення запису новою інформацією. Якщо ці команди додати до команд навігації, то ми отримуюмо найбільш популярний у кваліфікованих користувачів режим СУБД - Browse. Слід тільки пам'ятати, що режимом Browse зручно користуватися при його окремому виклику з головного меню або з командного вікна. А в програмах, створених на FoxPro, найчастіше використовують всеж-таки потрібні команди навігації.

Оскільки все вищезгадане відноситься тільки до однієї таблиці (*.dbf-файлу), перед користувачами СУБД у складних задачах автоматизації постає проблема одночасної навігації по декільком файлах. Для вирішення цієї проблеми в сучасних СУБД існують можливості по створенню реляційних зв'язків між *.dbf-файлами бази даних. Реляційні зв'язки мають односторонній напрямок від головного файлу до підлеглого (довідника). При встановленні реляційного зв'язку у FoxPro необхідно зафіксувати те поле з шифром головного файлу, по якому (див. лаб.роб. №2.2) має відшукуватися запис підлеглого файлу. У найпростішому випадку в полі з шифром головного файлу зберігається номер потрібного запису підлеглого файлу, але, якщо підлеглий файл проіндексовано (поняття індексації розглянуто в лаб.роб. №2.8) по такому ж самому власному полю з шифром, то реляційний зв'язок підтримується по значенню шифра.

Якщо реляційний зв'язок існує, то, відкривши одночасно два вікна Browse з головним та підлеглим файлом, можна побачити, що перехід на новий запис головного файлу супроводжується відповідним зміном запису підлеглого файлу. До речі, режим Browse дозволяє навіть в одному вікні побачити поля з різних баз, поєднаних реляційними або іншими зв'язками. Для цього необхідно запустити в командному вікні повну команду цього режиму у вигляді Browse Fields;... (детальніше про це можна дізнатися у розділі Browse HELP'у системи).

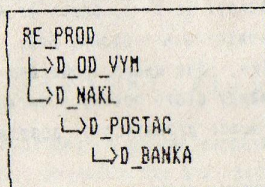
На прикладі когоспної ідальні можна зрозуміти, що кожен товар, занесений в реєстр, знаходиться тільки в одній накладній: ця ситуація відповідає реляційному зв'язку. Але не можна стверджувати навпаки - що в кожній накладній знаходиться тільки одна товарна позиція (хоча така ситуація не виключена). Оцей зворотний зв'язок у FoxPro, як і в інших СУБД, має назву один-до-багатьох, у нашому прикладі - кожній накладній відповідає визначе-

на множина товарів. Для реалізації цього типу зв'язку треба спочатку створити звичайний реляційний зв'язок в зворотньому напрямку (від підлеглого в попередньому випадку файлу до головного), а потім встановити режим "один-до-багатьох" та вказати в ньому поле з шифром головного файлу (останній має бути попередньо проіндексованим по цьому ж полю). Слід пам'ятати, що одночасно FoxPro не підтримує реляційний зв'язок та зворотній йому зв'язок типу "один-до-багатьох", оскільки циклічна реляція в СУБД неприпустима.

З М І С Т Р О Б О Т И

1. Для свого наскрізного варіанту бази даних запропонувати структуру реляційних зв'язків між головним файлом та файлами-довідниками та намалювати її. Вказати в примітках до рисунку ті поля (з шифрами), інформація в яких повинна забезпечувати потрібні реляційні зв'язки. Свій проект показати викладачеві та узгодити його з ним.

Розглянемо виконання роботи знову на прикладі колгоспної ідальні. Користуючись даними попередньої лабораторної роботи намалеємо наступну структуру потрібних файлів бази даних:



Мал.1. Ескіз реляційних зв'язків бази даних.

2. Створити в системі FoxPro відповідні власному мал.1. реляційні зв'язки в своїй базі даних за наступною методикою (Примітка: всі дії лабораторної роботи №2.3 виконуються або за допомогою миші, або за допомогою клавіш Tab, Enter та стрілочок):

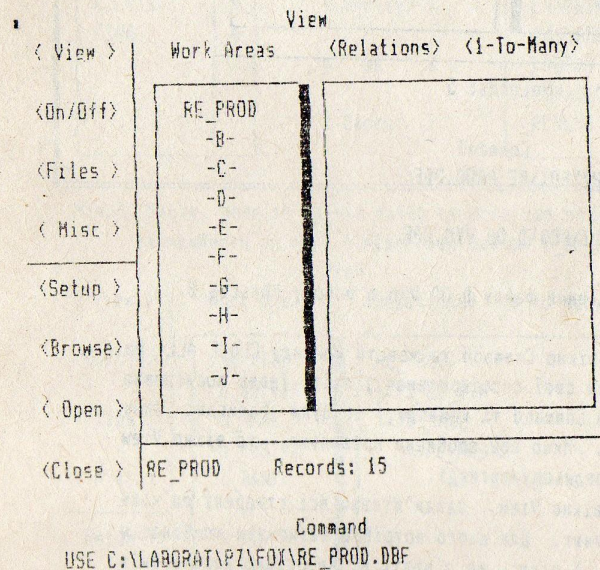
2.1. Відкрити вікно VIEW (режим перегляду бази даних) за допомогою WINDOW ! VIEW. Розташовувати вікна VIEW та COMMAND на екрані треба так, щоб вони не перекривали одне одного.

2.2. Стати на перший рядок лівого напіввікна VIEW з позначенням -A- (перша робоча область) та натиснути Enter. Вибрати

головний файл власної бази даних (кореневий на мал.1.). Занотувати у звіт ту команду, яка з'явиться у вікні COMMAND.

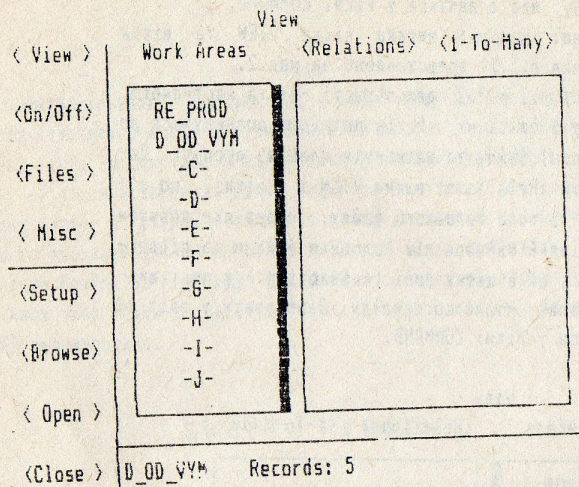
Приклад колгоспної ідальні: вигляд вікна VIEW та вікна COMMAND після виконання п.2.2. представлено на мал.2.

2.3. Виконати всі дії п.2.2. для області -B- та наступного, згідно мал.1, файлу бази даних. Потім потрібно повернутися до головного (попереднього) файлу та натиснути клавішу пропуск. Далі активізувати режим <Relations> вікна VIEW і у вікні, що з'явиться, вказати на те поле головного файлу, через яке повинен підтримуватися реляційний зв'язок між головним файлом та підлеглим. Мається на увазі, що в цьому полі (найчастіше - у полі шифру) має бути номер запису підлеглого файлу. Занотувати у звіт ту команду, яка з'явиться у вікні COMMAND.



Мал.2. Вигляд вікна VIEW та COMMAND.

Приклад створення реляційних зв'язків в базі даних колгоспної ідальні ілюструють мал.3,4,5.

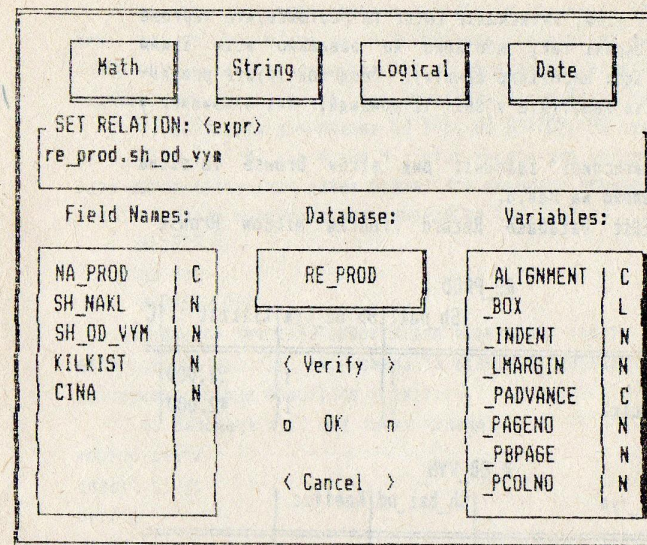


```
Command
USE C:\LABORAT\PI\FOX\RE_PROD.DBF
SELECT *
USE C:\LABORAT\PI\FOX\D_OD_VYM.DBF
```

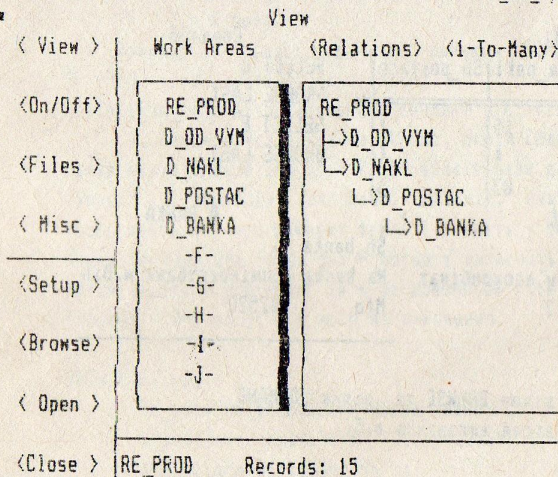
Мал.3. Встановлення файлу D_OD_VYM в робочу область В.

3. Перейти у вікно Command та ввести команду CLOSE ALL. Що відбулося? Записати свої спостереження у звіт. Тепер послідовно введіть через вікно Command ті команди, що були записані Вами при виконанні п.2. Якщо все зроблено правильно, то вікно View відновиться у попередньому вигляді.

4. Закрити вікно View, запам'ятавши всі створені зв'язки між файлами бази даних. Для цього потрібно натиснути комбінацію клавіш <Ctrl>+<W>. У вікні, що з'явиться, необхідно надати ім'я новоствореному файлу з розширенням *.VUE (<ім'я>.VUE). Спробуйте відновити режим VIEW через File ! Open головного меню, або командою SET VIEW TO <ім'я>.VUE.



Мал.4. Вибір поля sh_od_vym файлу re_prod для встановлення деяцїйного зв'язку з підлеглим файлом d_od_vym.



Мал.5. Кінцевий вигляд вікна View з усіма потрібними деяцїйними зв'язками.

5. Відкрити на екрані вікна з режимом Browse для всіх файлів бази даних таким шляхом: станьте послідовно на всі файли у лівому напіввікні View, натискайте Enter та розташовуйте чергове вікно Browse на екрані так, щоб воно не заважало всім іншим (вікно Command також необхідно бачити). Продемонструйте результат викладачеві та занотуйте у звіт ті команди, які з'явилися у вікні Command.

Приклад колегоспної ідальні: вид вікон Browse та вікна Command представлено на мал.6.

System File Edit Database Record Program Window Browse

RE_PROD				
Na_prod	Sh_nakl	Sh_od_vym	Kilkist	C
Морква	1	1	54.000	
Ковбаса московська	2	1	42.000	

D_OD_VYM			
Shifr_od	Na_od_vym	Sh_baz_od	Koeffit
1	кілограм		0.000
2	штука		0.000

D_NAKL				Command
Sh_nakl	Data_nakl	Nom_nakl	Sh_postach	
2	03.03.97	15	1	SELECT D
3	03.03.97	4	4	BROWSE LAST
4	28.02.97	87	3	SELECT E
				BROWSE LAST

D_POSTAC		D_BANKA	
Sh_postach	1	Sh_bank	1
Na_post	Районний м'ясокомбінат	Na_bank	Промінвестбанк м.Він
Rozr_rah	124512	Mfo	302520
Sh_bank	1		

Мал.6. Вид вікон BROWSE та вікна COMMAND після виконання п.5.

6. Спробуйте пересуватися по записам головного файлу в режимі Browse. Що в цей час відбувається у підлеглих файлах? А тепер зробіть спробу пересуватися по записам підлеглого файлу у відповідному вікні Browse. Спостереження занотуйте у звіт.

7. Достежати дію команд GO TOP, GO BOTTOM, GO <номер запису>, SKIP <+N/-M> (де N або M - цілі числа) для головного файлу бази даних (результат можна подивитися командою Browse). Відповідні команди, наприклад для команди go top, такі:

```
SELECT A
go top
BROWSE LAST
```

Спробуйте теж саме зробити для підлеглого файлу. Бисновки щодо результатів дії цих команд та відмінностей від аналогічних дій режиму Browse занотуйте у звіт.

8. Виконати п.7. для таких команд:

```
append blank
insert blank
delete
pack
replace (ім'я_поля/ім'я_значення_поля, ...)
Помітка: значення_поля повинно мати той самий тип даних, що і ім'я_поля.
```

Приклад колегоспної ідальні: команда REPLACE для файлу RE_PROD може мати наступний вигляд:

```
replace Na_prod with "Апельсини бочками", Kilkist with 20
```

9. Закрити всі вікна на екрані, окрім Command, та створити єдине вікно Browse для всіх *.dbf-файлів бази даних по типу наступного (з прикладу колегоспної ідальні). Нижчеказану команду треба набрати у програмному файлі (зайдіть у опції File | New, помітуйте клавішну Spras тип Program і натисніть на кнопку <OK>), потім надати *.prg-файлу ім'я за допомогою File | Save 1, через Program | Do, запустити його на виконання.

BROWSE FIELDS :

```
re_prod.Na_prod:15:h="Назва товару";
d_postac.Na_post:15:h="Постачальник";
d_nakl.Nom_nakl:2:h="На ";
d_nakl.Data_nakl:h="квдана";
re_prod.Kilkist:7:n="Кількість";
```



```
d_od_vym.na_od_vym:8:h='Од.вим.',;
re_prod.cina:6:h='Cina',;
suma=re_prod.kilkist*re_prod.cina:p='99999.99':h=' Сума ';
TITLE ' Зведені дані по товарам когоспної ідальні '
```

Примітка: програмні модулі краєм всього спочатку перевірити в "рідній" базі даних когоспної ідальні, файли якої відкрито для вільного перегляду та використання у розділі мережі Z:\NETOD\VPZ\EXAMPLE, а потім, на їх основі, створювати власні програми.

10. Аналогічно п.9. створити програмний файл для демонстрації зв'язку один-для-багатьох в єдиному вікні Browse. Методику створення індексного файлу, якого вимагає цей тип зв'язків, для вже відкритого *.dbf-файлу приведено в п.6. лаб.роботи N2.8.

Приклад програми для бази даних когоспної ідальні:

```
CLEAR
CLOSE ALL
SELECT A
USE D_NAKL.DBF
SELECT B
USE RE_PROD.DBF
SELECT A
* активізація індексного файлу по тегу (полю) SH_NAKL
SET ORDER TO TAG SH_NAKL OF RE_PROD.CDX IN RE_PROD
* створення реляційного зв'язку
SET RELATION TO SH_NAKL INTO RE_PROD ADDITIVE
* створення (накладення зверху) зв'язку типу один-до-багатьох
SET SKIP TO RE_PROD
BROWSE FIELDS ;
    d_nakl.nom_nakl:2:h='На',;
    d_nakl.data_nakl:h='кладна',;
    re_prod.na_prod:15:h='Назва товару',;
    re_prod.kilkist:9:h='Кількість',;
    re_prod.cina:6:h='Cina',;
    suma=re_prod.kilkist*re_prod.cina:p='99999.99':h=' Сума ',;
    z_summa=sl(sh_nakl):p='99999.99':h='Зар.сума ';
    TITLE ' Зведені дані по накладним когоспної ідальні '
```

* функція для визначення загальної суми по накладній

```
FUNCTION sl
parameters shifr
SUM re_prod.kilkist*re_prod.cina FOR shifr=re_prod.sh_nakl TO sl
return sl
```

Результати виконання пп.9 та 10 продемонструвати викладачеві, а свої враження від роботи програм по створенню складних режимів Browse занотувати у звіт.

В М І С Т З В І Т У

а) В звітті навести:

- ескіз системи зв'язків між файлами бази даних;
- послідовність команд створення реляційних зв'язків для власної бази даних;
- враження та висновки, що відповідають пп. 3,5,6,7,8,9,10 змісту роботи.

б) В своєму розділі мережі (електронний вигляд) мають бути:

- *.dbf-файл власної бази даних разом з файлом реляційних зв'язків *.vue;
- *.prg-файл для демонстрації складних режимів Browse власної бази даних згідно пп.9 та 10 змісту роботи.

П И Т А Н Н Я Д Л Я С А М О К О Н Т Р О Л Ю

1. Що таке реляційний зв'язок у базі даних?
2. Яка методика створення *.VUE-файлу з реляційними зв'язками в системі FoxPro?
3. Які команди відповідають створенню реляційних зв'язків в базах даних?
4. Як зміниться поведінка підлеглого файлу при накладенні на нього реляційного зв'язку?
5. Як відкрити існуючий *.VUE-файл різними способами?
6. Як занести інформацію в базу даних в командному режимі?
7. Які основні команди навігації по базі даних Ви знаєте?

8. Які можливості надає команда REPLACE та які Ви знаєте варіанти її використання?
9. Що роблять команди DELETE та PACK? Як помічаються в режимі BROWSE записи для знищення? Що відбувається з ними після команд PACK?
10. Як створити в базі даних зв'язок типу один-до-багатьох? Які команди відповідають цьому процесу?
11. Які можливості команди Browse Ви знаєте? Як показати в одному вікні Browse поля різних *.dbf-файлів, поєднаних зв'язками?

ЛАБОРАТОРНА РОБОТА №2.4

ТЕМА: СТВОРЕННЯ ГОЛОВНОГО МЕНЮ ДОДАТКУ ДО БАЗИ ДАНИХ ЗА ДОПОМОГОЮ ГЕНЕРАТОРА МЕНЮ (6 год.).

МЕТА РОБОТИ: ознайомлення з методикою створення *.prg-файлів в системі FoxPro та основними принципами побудови головного меню додатку; набуття навичок генерації меню в програмні файли та компіляції їх у файл, що виконується.

СТИСЛІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Генератор меню СУБД FoxPro, як і генератор екранів, відноситься до навічуд популярних тепер засобів "візуального програмування". Концепція "візуального програмування", на відміну від арсеналу традиційних мов програмування, дозволяє дуже швидко створювати потрібні елементи інтерфейсу шляхом їх формальної специфікації та простого розміщення на екрані комп'ютера. Іншими словами, програмні генератори, в тому числі і генератор меню, діють подібно до дитячого конструктора з тією різницею, що в результаті можна отримати не іграшкову будівлю, а щось на зразок порівняно простого макету меню або екрану. Цікаво відмітити, що власне головне меню СУБД FoxPro створено за допомогою свого ж генератора, тому всі системні функції головного меню FoxPro можна використовувати при розробці потрібних користувачеві меню.

Однак стриманий макет є лише першим етапом циклу програмування в СУБД FoxPro. Велими важливим вважається другий етап - генерація власне програмного файлу (тексту програми на мові FoxPro). Саме сгенерована програма в подальшому запускається на виконання, де проявляються її недоліки та помилки. Цей процес належить до третього етапу циклу програмування - аналізу програми та знаходження в ній синтаксичних та семантичних помилок. Виправити знайдені помилки можна двома наступними способами:

- використовувати сгенеровану програму як звичайну, тобто її можна змінювати, додавати потрібні команди або знищувати помилкові, вирізати з неї куски та вставляти в інші програми, тощо;
- замикнувши цикл програмування переходом на перший етап, а саме, виправивши помилкові дії програми в генераторі, створити новий макет і т.д. (цей спосіб використовується найчастіше).

Треба визнати, що концепція "візуального програмування", не дивлячись на її величезну популярність серед непрофесійних та початкових програмістів, нічого принципово нового не привнесла. Користування генераторами не набагато відрізняється від користування власними бібліотеками для створення об'єктів інтерфейсу на тому ж, наприклад, Турбо-Паскалі. Так, звичайно, ці бібліотеки необхідно спершу створити, але при цьому програміст набуває того досвіду, який потім дуже знадобиться для об'єднання тих же програмних кодів у єдину систему. Фактично, програмний цикл в СУБД відрізняється від програмного циклу Турбо-систем лише додатковим етапом створення та генерації макету меню або екрану. Тому, все-таки, найголовнішим у розробці власних додатків до бази даних є розуміння принципів програмування в СУБД та дії основних команд, а також навички відлагодження програм.

Головне меню програмного засобу відображає його функціональну структуру, тобто всі ті можливості, які закладено в програму її розробником. Для головного меню сучасних програмних засобів існує спеціальний стандарт текстового або графічного інтерфейсу ANSI. Згідно цього стандарту опції головного меню повинні бути розташованими у верхній лінійці екрану, підменю мають випадати донизу з "материнської" опції. Кількість та ступень вкладеності підменю визначається виключно функціями програми та системним мисленням програміста (як об'єднуються функції у групи та підгрупи).

Прийнято розрізняти опції меню верхнього рівня, активізація

яких призводить до появи підменю, та опції нижнього рівня, які запускатимуть конкретні задачі. Останні називаються також екранами, оскільки активізація відповідних опцій призводить до появи нової "картинки" на екрані монітора. В додатках до бази даних екрани забезпечують введення та виведення інформації, пошук та сортування даних, виконання необхідних обчислень, тощо. Таким чином, створення головного меню програмного засобу представляє собою процес проектування необхідних екранів та послідовності їх виклику за допомогою системи меню/підменю.

З точки зору користувача (замовника) програмного засобу найважливішим є вхід та вихід системи: які дані та з яких документів отримує додаток до бази даних і яка інформація та в якому вигляді виникає в результаті роботи програми. Тому в економічних та бухгалтерських додатках, як правило, в головному меню існують опції "Вхідні документи" та "Вихідні документи". Довідкову інформацію, яку, до речі, теж можна вважати вхідною, розміщують в опції "Довідники"; сервісні (додаткові) функції програми можна занести в опцію "Сервіс". Важливо не забути про опцію "Вихід", в якій необхідно забезпечити коректне закінчення роботи програмного засобу.

Розглянемо процес створення системи меню для додатку до бази даних на прикладі вже знайомої Вам колегоспної ідальні.

ЕСКІЗ СИСТЕМИ МЕНЮ

Головне меню додатку має наступний вигляд:

Вхідні документи Вихідні документи Довідники Сервіс Вихід

Опція "Вхідні документи" викликає наступне підменю, що випадає:

Накладні
Платіжні.вимоги

Ці дві опції повинні викликати відповідні екрани для введення інформації.

Опція "Вихідні документи" викликає наступне підменю, що випадає:

Реєстр накладних
Реєстр товарів
Звіт про фін.результати
Платіжні доручення

Перелічені опції повинні викликати відповідні екрани для виведення бухгалтерської інформації.

Опція "Довідники" викликає наступне підменю, що випадає:

Одиниці виміру
Банки
Постачальники
Накладні

Всі ці опції повинні викликати відповідні *.dbf-файли в режимі Browse (D_OD_VUM, D_BANKA, D_POSTAC, D_NAKL).

Опція "Сервіс" з навчальною метою демонструє так звані системні функції СУБД FoxPro, викликаючи наступне підменю, що випадає:

Калькулятор
Календар/записник
Дозвілля
Додаткові функції

В свою чергу, опція "Сервіс : Дозвілля" викликає власне підменю:

Гра 1
Гра 2
Гра 3
Поштова скринька

Будемо вважати, що опції "Календар/записник" та "Калькулятор" повинні викликати відповідні задачі-екрани, що вбудовані в систему FoxPro (в опції System). Опція "Гра 1" також має викликати стандартну гру FoxPro (System : Puzzle). Опція "Гра 2" запускає гру Coloboc.exe, яка знаходиться в каталозі z:\foxpro\.. Опція "Гра 3" запускає тренажер клавіатури z:\school\instruct.bat Опція "Поштова скринька" повинна викликати в редакторі Лексикон

файл Post.txt, що знаходиться на диску C:\. В свою чергу, опція "Сервіс ! Додаткові функції" викликає власне підменю:

Пошук
Сортування
Накладання фільтру

Всі ці опції повинні викликати відповідні екрани для забезпечення позначених функцій.

Опція Вихід викликає наступне підменю, що випадає:

В систему FoxPro
В систему DOS (дорослий)

Останні опції підменю повинні забезпечувати коректний вихід з додатку згідно з позначенням.

З М І С Т Р О Б О Т И

1. Для свого варіанту додатку до бази даних розробити ескіз головного меню аналогічно до наведеного в стислих теоретичних відомостях прикладу колегоспної ідальні. Ескіз узгодити з викладачем та занотувати у звіт.

2. Відкрити новий файл в генераторі меню: File ! New (вказати пропуском у стовпчику вибору (!) menu). Як і в деяких інших режимах (наприклад, Browse) у головному меню справа з'явиться нова опція Menu. Знайти в ньому і запустити режим Quick Menu.

3. Подивитися як створено системне меню FoxPro. Дослідити, в тому числі з використанням відповідного HELPу, та занотувати в звіт відповіді на наступні питання:

а) які можливі дії можуть використовуватись після активізації вибраної опції меню та в чому їх відмінності (Submenu, Bar, Proc, Command)?

б) як в меню можна створити "гарячі" клавіші та написати їх у відповідній опції?

в) яким чином задаються системні функції СУБД вищого та нижчого рівнів?

4. За допомогою клавіші Delete або активізації мишкою

відповідної кнопки на екрані знизити всі опції системного меню окрім Window. Через File ! Save as... записати файл меню під ім'ям m1.mnx. Створити програмний файл m1.prg за допомогою Program ! Generate. Відкрити отриманий файл m1.prg в іншому вікні через File ! Open (вказати пропуском у стовпчику вибору (!) program). Уважно передивитися текст сгенерованого програмного файлу та занотувати у звіт його структуру у вигляді блок-схеми алгоритму.

5. Через Program ! Do запустити файл m1.prg на виконання. Дослідити, як працює тепер система FoxPro. Занотувати у звіт висновок щодо невідновлюваності системного меню.

6. Вийти з системи FoxPro та знову запустити її. Ввести у вікні Command команду: push menu _sysmenu. Виконати знову дії п.5. Тепер ввести команду: pop menu _sysmenu. Висновки щодо процедури відновлення системного меню занотувати в звіт.

Примітка: команда Push заносить інформацію в стек оперативної пам'яті, а команда Pop виймає її з стека. Детальніше про це можна дізнатися в HELPі FoxPro.

7. Виконати ще раз дії п.2. Виконати дії п.4 так, щоб в меню лишилися опції System та Database. Записати програму меню у файл m2.mnx, а потім сгенерувати у файл m2.prg. Запустити файл m2.prg на виконання, дослідити роботу новоствореного меню та занотувати в звіт відповіді на наступні питання:

а) чи можна виконувати стандартні операції з вікнами меню або відповідними комбінаціями клавіш (відповідь обґрунтувати)?

б) що відбувається при відкритті режиму browse?

в) відповісти на питання б), попередньо давши у вікні Command команду: Set view to < назва свого *.vue - файлу >.

8. Перейти у вікно Command та дослідити дію наступних команд:

```
! coloboc.exe  
run coloboc.exe  
! z:\lexicon\lexicon  
run z:\foxpro\coloboc
```

Висновки щодо виклику програм з DOSа занотувати в звіт.

9. Перезапустити систему FoxPro. Виконати дії п.2. За допомогою клавіш Delete та Insert (для вставки опцій меню) лишити з системного меню тільки опції Windows і додати всі інші опції власного меню згідно ескізу, зробленому в п.1. Надати ім'я файлу меню m_menu.mnx, сгенерувати його у файл m_menu.prg та проде-

монструвати дім меню власного додатку викладачу.

Примітка: головне меню вашого додатку до бази даних має задовільняти наступним вимогам:

1. Всі опції меню та підменю зроблені на українській мові і мають "гарячі" клавіші для швидкого їх виклику.
2. Забезпечується весь сервіс роботи з вікнами.
3. Забезпечується коректний вихід в DOS та в систему FoxPro.
4. Автоматично завантажується свій *.VUE- файл.
5. Її опції підменю, які ще не мають потрібних екранів, просто не виконуються.

Приклад колгоспної ідальні. Головне меню зроблено згідно ескізу, що приведено в стислих теоретичних відомостях (була додана тільки опція "Вікно" як перейменована стандартна опція системи меню Window). Процес створення файлу m_menu.mnx ілюструється наступними екранами, які розміщено в мал.7-21.

System File Edit Database Record Program Window Menu

M_MENU.MNX

Prompt	Result	Options
\<Вхідні докуме	Submenu	< Edit > [X]
B\<вхідні докум	Submenu	< Edit > [X]
\<Довідники	Submenu	< Edit > [X]
\<Сервіс	Submenu	< Edit > [X]
Вих\<ід	Submenu	< Edit > [X]
Ві\<кно	Submenu	< Edit > [X]
		Menu Bar
		< Try it >
		Item
		< Insert >
		< Delete >

Мал.7. Вид головного меню в генераторі меню.

General Options

Procedure:

< Edit... >

o OK n

< Cancel >

Location:

Menu Code:

() Replace

() Append

() Before

() After

[X] Setup...

[] Cleanup...

[X] Mark...

Мал.8. Вікно General Options (викликається за допомогою Menu ; General Options; Setup.. позначити пропуском або мишею).

System File Edit Database Record Program Window
M_MENU.MNX
M_MENU Setup

SET VIEW TO KOLG_ID.VUE
PUSH MENU _MSYSMENU

Мал.9. Команди, що занесено у Setup.

Options

Comment:

o OK n

< Cancel >

[X] Shortcut...

[] Mark...

[] Skip For...

[] Pad Name...

Мал.10. Вигляд вікна для Options (позначити пропуском або мишею опцію Shortcut - задання "гарячої" клавіші).

Увага: в опції [] Pad Name... немає хрестика, що показує на відсутність системної функції вищого рівня: якщо після переробки системного меню хрестик є, необхідно зняти його так: мишею натиснути на нього, а потім - Tab, Del, Tab і Enter.

Key definition

Key Label: ALT+C

Key Text: ALT+C

Мал.11.Задання "гарячої" клавіші для опції "Сервіс".
System File Edit Database Record Program Window Menu

Prompt	Result	Options
\\Калькулятор Bar #	_MST_CALCUL [X]	_RJ00T9JK2 Try it Insert Delete
Календар/\\Запис Bar #	_MST_DIARY [X]	
Дозвілля Submenu	< Edit > []	
Додаткові функції Submenu	< Edit > []	

Мал.12.Вигляд підменю для опції "Сервіс" (назва _RJ00T9JK2 надана системою автоматично) в генераторі меню.

Увага: в меню використані в режимі Bar # системні функції СУБД FoxPro _MST_CALCUL та _MST_DIARY, які діють точно так само, як і у системному меню FoxPro.

System File Edit Database Record Program Window Menu

Prompt	Result	Options
Гра 1 Bar #	_MST_PUZZL []	_RJ00VORHS Try it
Гра 2 Command	run z:\foxpro\c []	
Гра 3 Command	run z:\school\p []	
Поштова скринька Command	! z:\lexicon\le []	

Item

Мал.13.Вигляд підменю "Сервіс ! Дозвілля" в генераторі меню.

System File Edit Database Record Program Window Menu

Prompt	Result	Options
Пошук Command	[]	_RJ00V5VIY Try it
Сортування Command	[]	
Накладання фільм Command	[]	

Item

Мал.14.Вигляд підменю "Сервіс ! Додаткові функції" в генераторі меню.

System File Edit Database Record Program Window Menu

Prompt	Result	Options
Одиниці виміру	Proc. < Edit > []	RJ00TA4LN
\-	Bar # _MFI_SP100 []	
Банки	Proc. < Edit > []	< Try it >
Постачальники	Proc. < Edit > []	
\-	Bar # _MFI_SP200 []	
Накладні	Proc. < Edit > []	Item < Insert > < Delete >

Мал.15.Вигляд підменю "Довідники" в генераторі меню.

Увага: в меню використані в режимі Bar # системні функції СУБД FoxPro _MFI_SP100 та _MFI_SP200, які забезпечують горизонтальний поділ між опціями меню.

System File Edit Database Record Program Window Menu

M_MENU.MNX
M_MENU Одиниці виміру Procedure

select b
browse

Мал.16.Вигляд вікна з процедурою для опції "Одиниці виміру".

System File Edit Database Record Program Window Menu

Prompt	Result	Options
Накладні	Command [X]	RJ00TAL03
\-	Bar # _MED_SP100 []	
Платіжн.вимоги	Command [X]	< Try it >
\-	Bar # _MED_SP200 []	Item < Insert > < Delete >

Мал.17.Вигляд підменю "Вхідні документи" в генераторі меню.

System File Edit Database Record Program Window Menu

Prompt	Result	Options
Реєстр накладні	Command []	RJ00TB6HN
\-	Bar # _MDA_SP100 []	
Реєстр товарів	Command []	
\-	Bar # _MDA_SP200 []	< Try it >
Звіт про фін.ре	Command []	
\-	Bar # _MDA_SP300 []	
Платіжні доруче	Command []	Item < Insert > < Delete >

Мал.18.Вигляд підменю "Вихідні документи" в генераторі меню.

System File Edit Database Record Program Window Menu

M_MENU.MNX
Prompt Result Options

В систему FoxPr	Proc. < Edit > [X]	RJ00TCZXM
\-	Bar # _MPR_SP100 []	
В систему DOS (	Command quit []	< Try it >
		Item < Insert > < Delete >

Мал.19.Вигляд підменю "Вихід" в генераторі меню.

System File Edit Database Record Program Window Menu

M_MENU.MNX
M_MENU В систему FoxPr Procedure

clear all
clear

pop menu _msysmenu

Мал.20. Вигляд вікна з процедурою виходу "В систему FoxPro".

System File Edit Database Record Program Window Menu

M_MENU.MNX

Prompt	Result	Options
\<Hide	Bar #	MWI_HIDE []
\<Hide All	Bar #	MWI_HIDEA []
Sh\<om All	Bar #	MWI_SHOWA []
Clea\<r	Bar #	MWI_CLEAR []
\-	Bar #	MWI_SP100 []
\<Move	Bar #	MWI_MOVE [X]
\<Size	Bar #	MWI_SIZE [X]
\<Zoom	Bar #	MWI_ZOOM [X]
Z\<oom	Bar #	MWI_MIN [X]
\<Cycle	Bar #	MWI_ROTAT [X]

RJ00TDFIS

< Try it >

Item
< Insert >

< Delete >

Мал.21.Вигляд підменю "Вікно" в генераторі меню.

Увага: в меню використані в режимі Bar # ті системні функції СУБД FoxPro, які відповідають опції Window системного меню FoxPro.

В М І С Т З В І Т У

Звіт з лабораторної роботи обов'язково повинен містити:

- відповіді на питання по всім пунктам змісту роботи окрім п.9.;
- файли m1.mpr, m2.mpr, m_menu.mpr (в електронному вигляді).

П И Т А Н Н Я Д Л Я С А М О К О Н Т Р О Л Я

1. Якими принципами необхідно керуватися при побудові головного меню програмного засобу?
2. Що необхідно визначити для того, щоб намалювати ескіз головного меню?
3. Яким чином вставити DOSівські команди в меню FoxPro?
4. Як можна вкличити стандартні опції меню FoxPro у власне меню додатку?
5. Що можна вставити в процедури Setup та Cleanup меню та для чого вони потрібні?

6. Як досягти стандартних можливостей (пересування по екрану, зміна розміру) для вікон Browse, що викликаються з меню додатку?
7. Як зробити виклик *.uve-файлу з меню? Для чого це потрібно?
8. Які команди забезпечують відновлення системного меню після дії меню додатку?
9. Як встановити "гарячі" клавіші для меню додатку? Навіщо це потрібно?
10. Як згенерувати програмний файл з меню додатку, що створено в генераторі меню? Яке розширення він має?
11. Яка структура файлу з програмою для меню?

Л А Б О Р А Т О Р Н А Р О Б О Т А № 2.5

ТЕМА: ПРИНЦИПИ ПОБУДОВИ ГЕНЕРАТОРА ЕКРАНІВ СУБД FOXPRO ТА МЕТОДИКА ЙОГО ВИКОРИСТАННЯ (4 год.).

МЕТА РОБОТИ: ознайомлення з методикою створення *.srg-файлів в системі FoxPro та основними можливостями генератора екранів; набуття навичок генерації екранів з елементами інтерфейсу в програмні файли та компіляції їх у файли, що виконуються.

С Т И С Л І Т Е О Р Е Т И Ч Н І В І Д О М О С Т І

Генератор екранів СУБД FoxPro, що був у свій час одним з перших програмних засобів "візуального програмування", і дотепер відзначається неабиякими зручностями в користуванні. Генератор базується на широко розповсюдженому нині понятті екранного об'єкту та має повний асортимент можливостей по маніпулюванню об'єктами інтерфейсу різних типів. Оскільки сама система FoxPro використовує при спілкуванні з користувачем ті ж самі об'єкти, то і екрани, створені генератором, зовні схожі на інтерфейс FoxPro та інтуїтивно зрозумілі.

До екраних об'єктів відносять рамки, фрагменти тексту, поля баз даних (або складні вирази з цих полів), а також різного роду кнопки та перемикачі. При відкритті генератора екрану справа у

головному меню з'являється опція Screen. У ній представлені всі можливі об'єкти, які ми послідовно розглянемо.

1. Box (Рамка). Розміщує рамку в позиції курсора в вікні Screen Design (проектуювання екрана). Нова рамка мигтить, що вказує на те, що можна змінювати її розміри стрілочками. На екрані виникає діалог Box для вибору чинників рамки.

2. Field... (Поля). Викликає діалог Screen Field (Поле екрана), за допомогою якого можна розмістити на екрані поле бази даних або побудувати складний вираз з полів.

3. Text (Текст). Дозволяє створювати і редагувати текстові об'єкти.

4. Push Button... (Індикатор). Викликає діалог Push Button, який задає чинники для кнопки, що діє під час її активізації.

5. Radio Button... (Груповий індикатор). Викликає діалог Radio Button, який задає чинники для ряду з кількох кнопок, причому активною у будь-який час може бути лише одна з них.

6. Check Box... (Контрольне вікно). Викликає діалог Check Box для визначення чинників кнопки, яка при активізації дозволяє закрити або відкрити визначену опцію.

7. Popup... (Вертикальне меню). Викликає діалог Popup, який задає чинники для активізації вертикального меню з полів бази даних, елементів масиву або інших допустимих виразів.

8. List... (Список). Викликає діалог List, який задає чинники для активізації списку з полів бази даних або інших допустимих виразів.

9. Invisible Button... (Невидимий індикатор). Цей діалог дозволяє створити невидиму кнопку, яка при активізації може виконати потрібні для екрану дії.

Детальніше про всі ці екранні об'єкти та відповідні їм діалог можна дізнатися в HELP-і системи, натиснувши клавішу F1 під час перебування в потрібному режимі головного меню. Характерною ознакою FoxPro є те, що для кожного з перерахованих вище об'єктів генератор екранів забезпечує виконання наступних функцій:

- створення об'єкту : пересуньте курсор або показчик миші на обране місце екрану і виберіть потрібний тип об'єкта з меню;
- пересування між об'єктами : натискайте кнопку миші на кожному об'єкті, або клавішу Tab або Shift+Tab;
- вибір (позначення) об'єкту : натисніть кнопку миші на об'єкті, або пересуньте курсор на об'єкт і натисніть клавішу

Spacebar (пропуск);

- вибір (позначення) множини об'єктів : натискайте Shift + кнопка миші або клавішу Shift+Spacebar (пропуск) на об'єктах, що повинні увійти у множину. Щоб відділити об'єкти, пересуньте вказівник миші за межі потрібних об'єктів і звільніть кнопку миші, або натискайте Spacebar (пропуск) до виходу за межі об'єктів, використовуючи клавіші-стрілки; натисніть Enter, коли визначите множину;

- відміна вибору об'єкта : Shift+кнопка миші або Shift + Spacebar (пропуск) на об'єкті;

- відміна вибору всіх об'єктів : натисніть кнопку миші або Spacebar (пропуск) за межами об'єктів;

- пересування об'єкта : натисніть кнопку миші на об'єкті і пересуньте його мишею, або виберіть об'єкт, пересуньте курсор за допомогою клавіш-стрілок та натисніть Enter;

- зміна довжини об'єкта : натисніть Ctrl+кнопку миші і пересуньте вказівник, або натисніть Ctrl+Spacebar (пропуск), перевизначить розмір за допомогою клавіш-стрілок та натисніть Enter;

- модифікування об'єкту : подвійне натискання кнопки миші на об'єкті, або пересуньте курсор на об'єкт і натисніть Enter;

- редагування текстового об'єкту : натисніть Ctrl+Spacebar (пропуск) на об'єкті, відредагуйте його та натисніть Enter після закінчення редагування.

Про інші опції режиму Screen, що стосуються операцій з екранними об'єктами, можна дізнатися у HELP-і по генератору екранів. Процес збереження екрану в файл з розширенням *.scx та наступна генерація його в програмний файл з розширенням *.spr, який вже можна виконувати, повністю співпадає з вже знайомими Вам аналогічними процесами генератору меню. Слід також відмітити, що в системі FoxPro існує можливість побудови "невидимого екрану", який відображає активну в поточний час базу даних у вигляді множини її полів. Такий екран хоча і є лише примітивним подобою Browse, але це вже програма, а не команда.

З М І С Т Р О Б О Т И

1. Увійти в систему FoxPro, відкрити вікно View та запус-

ти у вікні Command команду SET VIEW TO <ім'я>.VUE для відкриття свого 1.VUE-файлу. У вікні View стати на ту базу даних, яка є головною для першого з екранів, та натиснути клавішу пропуску.

Вид вікна Command для прикладу колгоспної ідальні показано на мал.22.

```
Command
clear
SET VIEW TO KOLG_ID.VUE
SELECT A
```

Мал.22. Вид вікна Command.

2. Відкрити новий файл в генераторі екранів: File ; New (вказати пропуском у стовпчику вибору (1) Screen). На екрані монітора буде відкрито спеціальне робоче вікно для створення нового екрану, а у головному меню праворуч з'явиться нова опція Screen. Знайти в ньому і запустити режим Quick Screen, у діалогі, що з'явиться, нічого не міняти та натиснути Enter.

Приклад колгоспної ідальні: після виконання п.2 генератор екранів має такий вигляд, як на мал.23.

```
System File Edit Database Record Program Window Screen
UNTITLED.SCX
R: 0 C: 0 | Move |
Na_prod 1: na_prod.....
Sh_nakl 2: s
Sh_od_vym 3:
Kilkist 4: kilkist
Cina 5: cina...
```

Мал.23. "Швидкий екран" для файлу Re_prod.dbf.

3. Подивитись, як створиться швидкий екран для відкритої в системі FoxPro бази даних. Врахувати, що найпростіший екран вміщує тільки два типи об'єктів - Field... та Text. Дослідити на цьому екрані, в тому числі з використанням стислих теоретичних відомостей або відповідного HELPу до режиму Quick Screen, та занотувати в звіт відповіді на часті питання:

а) як працює опція Screen ; Quick Screen, якщо в поточний момент її активізації не відкрита кодна з баз даних?

б) як записати на екрані текст та модифікувати його?
в) що можна врахати екранним об'єктом?
г) як позначати об'єкти (групи об'єктів) та знімати позначення?

д) як пересувати об'єкти (групи об'єктів) на екрані та змінювати їх розмір?

е) як знищувати об'єкти (групи об'єктів) на екрані?

ж) як модифікувати екранні об'єкти?

4. Через File ; Save as... записати файл екрану під іменем S_QUICK.scx. Створити програмний файл S_QUICK.spr за допомогою Program ; Generate. Відкрити отриманий файл S_QUICK.spr в іншому вікні через File ; Open (вказати пропуском у стовпчику вибіра (1) program). (Якщо передивитися текст сгенерованого програмного файлу та занотувати у звіт його структуру у вигляді блок-схеми алгоритму.

Приклад колгоспної ідальні. Текст програмного файлу S_QUICK.spr (з коментарями авторів методичних вказівок):

*REGION 0 && розділ з початковими деклараціями для роботи з && пам'яттю:

REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"

SET TALK OFF

m.talkstat = "ON"

ELSE

m.talkstat = "OFF"

ENDIF

m.compstat = SET("COMPATIBLE")

SET COMPATIBLE FOXPLUS

m.currarea = SELECT()

!

!

!

!

!

!

*REGION 1

&& розділ з операторами введення-виведення даних:

@ 0.0 SAY "Na_prod"

@ 0.11 GET re_prod.na_prod ;

SIZE 1.30 : && розмір поля GET:

S_QUICK Screen Layout


```

DEFAULT " "  && зміст поля GET по змовчуванню:
@ 1,0 SAY "Sh_nakl"
@ 1,11 GET re_prod.sh_nakl ;
    SIZE 1,4 ;
    DEFAULT 0
@ 2,0 SAY "Sh_od_vym"
@ 2,11 GET re_prod.sh_od_vym ;
    SIZE 1,2 ;
    DEFAULT 0
@ 3,0 SAY "Kilkist"
@ 3,11 GET re_prod.kilkist ;
    SIZE 1,10 ;
    DEFAULT 0
@ 4,0 SAY "Cina"
@ 4,11 GET re_prod.cina ;
    SIZE 1,10 ;
    DEFAULT 0

READ CYCLE && оператор циклічного очікування введення інформації
&& з клавіатури:
SELECT (a.currareal)

#REGION 0      && розділ з заключною ініціалізацією:
IF m.talkstat = "ON"
    SET TALK ON
ENDIF
IF m.comostat = "ON"
    SET COMPATIBLE ON
ENDIF

```

5. Через Program ! До запуску файлу S_QUICK.spr на виконання (при цьому Ви повинні бачити вікно View). Дослідити, як працює сгенерований екран в системі FoxPro. Занотувати у звіт відповіді на наступні питання:

- що відбувається у вікні View при завантаженні екрану S_QUICK.spr ?
- як інформація з'явилася на екрані монітора ?
- як вийти з екрану ?
- що з'явилось на екрані монітора після виходу з екрана

S_QUICK.spr ?

Приклад когоспної ідальні (після виконання п.5.) ілюструється мал.24.

Na_prod	картопля
Sh_nakl	1
Sh_od_vym	1
Kilkist	1.000
Cina	3.00

Мал.24. Результат роботи "швидкого екрану" іочевидно, що даний екран примітивний, оскільки показує тільки один запис бази re_prod, у тому числі досить такі незрозумілі цифри накладних та одиниць виміру).

5. Відкрити файл з екраном S_QUICK.SCX за допомогою команди MODIFY SCREEN S_QUICK.SCX або через File ! Open головного меню, вказавши пропуском у стовпчику вибору (1) screen. Метод Ваших подальших дій має бути дослідження на цьому робочому екрані процесів створення основних елементів інтерфейсу, закладених в генераторі екранів FoxPro. В процесі роботи користуйтеся стислими теоретичними відомостями або звертайтеся за довідками до відповідного HELPу системи FoxPro.

Примітка: особливості складніших загальних режимів Screen Layout... та Open All Snippets будуть детально розглянуті в наступній лабораторній роботі.

5.1 Зробити на екрані рамку, ставши маркером на її лівий верхній вугол, з потім Screen Box або B. Далі встановити та зафіксувати правий нижній вугол рамки. У діалозу, що з'явиться можна нічого не змінювати.

Виконати п.4 та знайти у файлі S_QUICK.spr відмінності від його попередньої версії. Запустити файл S_QUICK.spr на виконання та проаналізувати його роботу. Висновки занотувати у звіт.

5.2 Встановити на екрані кнопку, що натискається: Screen ! Push Button... або H. Зафіксувати такий вид діалогу, що з'явиться, який представлено на мал.25.

Push Button Prompts:

Можлива дія 1
Можлива дія 2
Можлива дія 3

() Horizontal (#) Vertical
[] Terminating (Spacing...)

Variable:

< Choose... > v1

Options:

[] When... [] Comment...
[X] Valid... [] Disabled
[] Message...

o OK n < Cancel >

Мал.25. Діалог для встановлення кнопки Push Button...

Змінна v1 показує на те, яка з кнопок обрана. Програмна функція Valid... яка починає виконуватися після натискання на кнопку, використовує значення v1 і має такий вигляд (мал.26.):

```

S_QUICK v1 VALID
DO CASE
CASE v1 = 1
? chr(7)
CASE v1 = 2
? chr(7)
? chr(7)
CASE v1 = 3
? chr(7)
? chr(7)
? chr(7)
ENDCASE

```

Мал.26. Програмна функція, яка виконується при натисканні кнопки Push Button... Коментар: системна команда ? chr(7) викликає короткий сигнал.

Виконати п.4 та знайти у файлі S_QUICK.sdr відмінності від

його попередньої версії. Запустити файл S_QUICK.sdr на виконання та проаналізувати його роботу. Висновки занотувати у звіт.

5.3 Створити кнопку вибору: Screen ! Radio Button... або "N. Зафіксуйте такий вид діалогу, що з'явиться (мал.27.):

Radio Button Prompts:

Такої
Чи такої
Або отакої

(#) Horizontal () Vertical

Variable:

< Choose... > v2

Options:

[] When... [] Comment...
[X] Valid... [] Disabled
[] Message...

Initial: Чи такої o OK n < Cancel >

Мал.27. Діалог для групи кнопок Radio Button...

Текст функції Valid... має такий вигляд (мал.28.):

```

S_QUICK v2 VALID
DO CASE
CASE v2=1
@ 5.0 to 18.75
CASE v2=2
@ 5.0 to 18.75 double
CASE v2=3
@ 5.0 to 18.75 panel
ENDCASE

```

Мал.28. Програмна функція, яка виконується при натисканні кнопки Radio Button... Коментар: ця функція перемальовує рамку на екрані трьома можливими варіантами (мається на увазі, що рамку саме з такими розмірами було створено в п.6.1).

Виконати п.4 та знайти у файлі S_QUICK.spr відмінності від його попередньої версії. Запустити файл S_QUICK.spr на виконання та проаналізувати його роботу. Висновки занотувати у звіт.

6.4 Створити кнопку бінарного вибору: Screen ! Check Box... або "X". Зафіксуйте такий вид діалогу, що з'явиться (мал.29.):

Мал.29. Діалог для створення кнопки Check Box...

Текст функції Valid... має такий вигляд (мал.30.):

```
S_QUICK v3 VALID
if v3=1
wait window "Так, я все зрозумів"
else
wait window "Я щось не зрозумів"
endif
```

Мал.30. Програмна функція кнопки Check Box... Коментар: системна функція wait window викидає на екран вікно з попередженням для користувача.

Виконати п.4 та знайти у файлі S_QUICK.spr відмінності від його попередньої версії. Запустити файл S_QUICK.spr на виконання та проаналізувати його роботу. Висновки занотувати у звіт.

6.5 Створити кнопку для меню, що впливає: Screen ! Popup... або "O". Зафіксуйте такий вид діалогу, що з'явиться (мал.31.):

Мал.31. Діалог для створення меню.

У цій кнопці використана опція [X] When..., в якій закладено функцію, що починає діяти, на відміну від опції Valid..., при переході на екранний об'єкт (тобто, перед можливим вибором, а не після). Текст функції When... має такий вигляд (мал.32.):

```
S_QUICK v4 WHEN
? chr(7)
```

Мал.32. Програмна функція When...

Виконати п.4 та знайти у файлі S_QUICK.spr відмінності від його попередньої версії. Запустити файл S_QUICK.spr на виконання та проаналізувати його роботу. Висновки занотувати у звіт.

6.6 Створити кнопку для меню у вигляді списку: Screen ! List... або "L". Зафіксуйте наступний вид діалогу, що з'явиться (мал.33.):

List Type:	Options:
☐ From Array ☐ From Popup ☐ Prompt Structure ☒ Prompt Field na_prod ☐ Prompt Files	☐ When... ☐ Comment... ☒ Valid... ☐ Disabled ☐ Message... ☐ 1st Element... ☐ Terminating ☐ # Elements...
Variable:	
< Choose... > v5	☐ OK ☐ Cancel

Мал.33. Діалог для створення списку List...

Текст функції Valid... має такий вигляд (мал.34.):

```

S_QUICK v5 VALID
wait window "Ви вибрали: " + alltrim(na_prod)
show gets     && показати нові значення полів (field)
return .t.    && коректний логічний вихід з функції
    
```

Мал.34. Програмна функція для списку List... Коментар: функція alltrim(na_prod) віддає значення рядкового поля na_prod без попередніх та кінцевих пропусків.

Виконати п.4 та знайти у файлі S_QUICK.spr відмінності від його попередньої версії. Запустити файл S_QUICK.spr на виконання та проаналізувати його роботу. Висновки занотувати у звіт.

6.7 Створити невидиму кнопку: Screen ! Inv. Button... або ^I. Зафіксувати такий вид діалогу, що з'явиться (мал.35.):

Invisible Buttons #: 1	
☐ Horizontal ☐ Terminating	☐ Vertical ☐ <Spacing...>
Variable:	
< Choose... > v6	☐ OK ☐ Cancel
Options:	
☐ When... ☐ Comment... ☒ Valid... ☐ Disabled ☐ Message...	

Мал.35. Діалог для створення невидимої кнопки.

Текст функції Valid... з навчальною метою має такий вигляд (мал.36.):

```

S_QUICK v6 VALID
wait window "Ви натиснули на невидиму кнопку - поздоровяємо !"
    
```

Мал.36. Програмна функція для невидимої кнопки.

Виконати п.4 та знайти у файлі S_QUICK.spr відмінності від його попередньої версії. Запустити файл S_QUICK.spr на виконання та проаналізувати його роботу. Висновки занотувати у звіт.

Приклад когоспної ідальні: в результаті всіх попередніх дій файл з екраном S_QUICK.SCX може мати такий вигляд (мал.37.):

System File Edit Database Record Program Window Screen
S_QUICK.SCX

R: 0 C: 0 | Move |
Na_prod 1: na_prod.....
Sh_nakl 2: s
Sh_od_vym 3:
Kilkist 4: kilkist
Cina 5: cina...

< Можлива дія 1 > () Такої (!) Чм такої () Або отакої
< Можлива дія 2 >
< Можлива дія 3 >
[] Все зрозуміло
Обізнаний

Мал.37. Кінцевий вигляд екрану S_QUICK.SCX.

Результати дії сгенерованого файлу S_QUICK.SPR можна побачити на мал.38.

Na_prod Ковбаса московська
Sh_nakl 2
Sh_od_vym 1
Kilkist 42.000
Cina 9.30

< Можлива дія 1 > () Такої (!) Чм такої () Або отакої
< Можлива дія 2 >
< Можлива дія 3 >
[X] Все зрозуміло
Майбутній босс

Мал.38. Результат роботи сгенерованого у файл S_QUICK.SPR екрану S_QUICK.SCX.

В М І С Т З В І Т У

- Звіт з лабораторної роботи обов'язково повинен вмішувати:
- розгорнуті відповіді на питання по тим пунктам змісту роботи, де є згадка про звіт;
 - файл S_QUICK.SCX та сгенерований на його основі файл S_QUICK.SPR (в електронному вигляді).

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Якими принципами необхідно керуватися при побудові екранів програмного засобу на основі генератору екранів?
2. Які екранні об'єкти існують в системі FoxPro?
3. Як стандартні операції з об'єктами забезпечує генератор екра-

нів FoxPro?

4. Як сгенерувати програмний файл, що викликає екран, створений в генераторі екранів? Яке розширення має така програма?
5. Яка структура файлу з програмом для виклику екрана?
6. Навіщо використовується змінна (Variable) при створенні кнопок на екрані?
7. Яке призначення програмної функції з опції Valid?
8. Яке призначення програмної функції з опції When?
9. Що визначається опцією Initial?

ЛАБОРАТОРНА РОБОТА N 2.6

ТЕМА: СТВОРЕННЯ СКЛАДНИХ ЕКРАНІВ ДЛЯ ДОДАТКУ ДО БАЗИ ДАНИХ ЗА ДОПОМОГОЮ ГЕНЕРАТОРА ЕКРАНІВ (6 год.).

МЕТА РОБОТИ: ознайомлення з методикою створення складних елементів інтерфейсу в системі FoxPro на прикладі власного додатку до бази даних; набуття навичок відлагодження програмних функцій, потрібних для коректної роботи екранів додатку.

СТИСЛІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Сгенеровані системою FoxPro екрани для складних задач автоматизації не можна вважати лише набором екранних об'єктів. Річ у тому, що всі об'єкти інтерфейсу повинні чітко взаємодіяти між собою в залежності від ситуації, що склалася на екрані, а також виконувати заздалегідь визначені функції. Тобто екранні об'єкти не є незалежними, бо мають між собою зв'язки для виконання закладених в екран функцій. У відтворенні потрібних зв'язків, яке можна досягти лише програмно, і полягає найбільша складність при розробці екранів додатку до власної бази даних.

В системі FoxPro всі програмні функції, що використовуються при розробці екрану, мають назву Snippet. На них якраз і покладено задачу забезпечення всіх необхідних дій екранних об'єктів і налагодження потрібних взаємозв'язків між ними. Оскільки змістом будь-якого Snippet'у завжди є програма, в системі FoxPro перед-

бачені можливості по окремій обробці таких програм.

Так в опції Screen головного меню вміщено режим Open all snippets, який відкриває для редагування в окремих вікнах всі програмні модулі, що були створені для поточного екрану в усіх екранних об'єктах.

Загальні параметри екрану, що генерується, закладено в опції Screen : Layout... (формат екрану). Треба чітко уявляти собі, що всі екранні об'єкти розташовуються на віртуальному піддоні екрану (аналогія - загрунтований холст картини), властивості якого і визначає опція Layout. Тут можна змінювати такі глобальні параметри поточного екрану, як його тип та розміри, заголовки, початкові та кінцеві установки для роботи екрану, тощо. Всі вони доступні та діють для всіх інших програмних кодів, які задіяні в об'єктах даного екрану. Конкретніше опція Screen : Layout... вміщує наступні можливості:

() Desktop (Дощка) - розміщує всі екранні об'єкти безпосередньо на всьому екрані монітора (режим "дощки");

() Window (Вікно) - виводить всі екранні об'єкти в спеціальне вікно.

В цьому випадку необхідно визначити наступні параметри вікна:

Name: - ім'я вікна (ім'я вікна генерується, якщо не визначено);
Title: - заголовок (текст, який з'являється в рядку заголовка вікна);

Footer: - нижній колонтитул (текст, який з'являється в нижній частині вікна);

Height: - висота вікна;

Width: - ширина вікна (Примітка: останні два параметри найкраще визначати безпосередньо в генераторі екрану, фіксуючи мимовільно правий нижній кут вікна);

Row: та Column: - визначають розмір керуючої області для розміщення об'єктів на "дощці";

[X] Center - центрування: якщо обрано, то вікно автоматично центрується на моніторі;

Type... - тип, викликає діалог Type, який вміщує наступні опції:

Type - вибір опції для призначення атрибутів і обмежувача у вікні згідно обраної опції. Вибір User робить доступними всі наступні опції цього діалога:

- ☐ Close - при виборі дозволяє користувачеві закрити вікно за допомогою мени, клавіатури або миші;
 - ☐ Float - при виборі дозволяє користувачеві пересунути вікно у будь-яку позицію екрана;
 - ☐ Zoom - при виборі дозволяє користувачеві збільшити вікно до розмірів екрану;
 - ☐ Shadow - при виборі вікно створюється із тінню;
 - ☐ Minimize - при виборі вікно може бути стиснуто до 1 рядку з 16 колонок;
- існує також можливість визначити один з типів обмежувача:
- ☐ None - вікно без обмежувача;
 - ☐ Single - кордон вікна - проста лінія;
 - ☐ Double - кордон вікна - подвійна лінія;
 - ☐ Panel - вікно має широкий кордон (як у системного вікна), але відсутні керуючі символи;
 - ☐ System - вікно має широкий кордон; керуючі символи видимі;
- для визначення кольорів (кольорових схем) елементів інтерфейсу існують опції:

- Primary
- дозволяє вказати за замовчуванням кольорові схеми для вікна і об'єктів, що знаходяться в ньому, за виключенням мени;

Popur

- Дуже важливими опціями режиму Screen ! Layout... вважаються:
- ☐ Setup... - якщо обрано, то відкривається вікно редагування для визначення програми установки (остання діє перед завантаженням екрану);
 - ☐ Cleanup & Procs... - якщо обрано, то відкривається вікно редагування для визначення коду очистки і процедур, що використовуються при роботі екрану.

В складних задачах можна використати також опції режиму READ (Читати), в яких визначити відповідні програми:

- ☐ Activate... (Активізувати для екрану)
- ☐ Show... (Показати перед дією екрану)
- ☐ Valid... (Значення - виконати при активізації екрану)
- ☐ When... (Коли - виконати перед дією екрану)
- ☐ Deactivate... (Відмінити для екрану)

Примітка: якщо обрано, то викликається діалог Code Snippet (даний відповідний HELP) для цих адрес READ; принципово програми цих опцій лише систематизують той програмний код, який замість них може бути вміщено в опції Setup... - різні варіанти закладення кодів роблять генератор екранів більш гнучким.

Дуже важливі функції - збереження, відтворення або очищення інформації про середовище FoxPro, в якому було створено екран, забезпечують наступні кнопки режиму Environment:

- < Save > < Restore > < Clear >
(Зберегти Відтворити Очистити)

Примітка: якщо під час першого створення екрану не була натиснута одна з цих кнопок, то система FoxPro спеціально перелічує щодо збереження інформації про середовище (які бази відкриті і в яких робочих областях (.VUE-файл), які зроблено установки типу Set, тощо).

Остання опція режиму Layout (якщо обрано) додає псевдонім бази даних до всіх полів, що зустрічаються з екрані:

- ☐ Add Alias - додати псевдонім.

А тепер розглянемо процес створення екранів, що відповідають системі мени, для додатку до бази даних на прикладі колгоспної ідальні. Для досягнення цієї мети безпосередньо в змісті роботи будуть наведені відповідні приклади дій та програмних модулів, що відповідають екрану та головному мени колгоспної ідальні (всі необхідні файли додатку до бази даних відкрито для зільного перегляду та використання у розділі черехі I:\METOD\PZ\EXAMPLE).

ЗМІСТ ДОДАТКІВ

1. Для свого варіанту додатку до бази даних розробити ескізи відповідних структурі головного мени екранів аналогічно до наведеного нижче прикладу колгоспної ідальні. Ескізи узгодити з викладачем та занотувати у звіті.

Поняма колгоспної ідальні ілюструє мал.39. та мал.40.

2. Дати у вікні Command команду Clear, користуючись набутими навичками та інформацією з стислих теоретичних відомостей, прикладом колгоспної ідальні та підказками HELP створити перший екран згідно з першим ескізом. 3.1. Назвати ім'я файлу екрана s_1.vue, збережувати його у файлі s_1.sor та продемонструвати дію

цього екрану власного додатку викладачу.

Примітка: кожний екран Вашого додатку до бази даних має задовільняти наступним вимогам:

1. Всі написи на екрані зроблені українською мовою.
2. Екран знаходиться у вікні, для якого забезпечується весь стандартний сервіс роботи з вікнами.
3. Забезпечується коректний виклик екрану з глобального меню додатку та очищення екрану монітора після закінчення роботи.
4. Викликається подвійне завантаження свого *.VUE-файлу після виклику екрану.
5. На екрані не має бути таких кнопок, які або не мають потрібних підпрограм та функцій або взагалі не виконуються.

Робота з накладними				
Накладна N		23	січень	
від		07.01.97		
(внутрішній збір		1)		
Постачальник КСП "Південнобузьке"				
кому колгоспний ідальні				
Назва товару	Кількість	Од.вимір.	Ціна	Сума
Картопля	5.00	тона	1.00	5.00
Буряк	50.00	штука	5.00	250.00
Морква	30.00	кілогр	2.50	75.00
Контейнери	5.00	штука	2.00	10.00
Загальна сума накладної:				340.00
(Нова накладна) < Пошук > () перегляд () редагування Tab - перехід між полями; Esc - Вихід				

Мал.39. Ескіз екрану для введення накладних
(відповідає опції Вихідні документи / Накладні).

Редагування даних товару				
Назва товару	Кількість	Од.вимір.	Ціна	Сума
Буряк	50.000	штука	5.00	250.00
< Додати рядок > < Знищити рядок > Tab - перехід між полями; Esc - Вихід				

Мал.40. Ескіз екрану для корегування рядка накладної
(відповідає виклику рядку для товару "Буряк" попереднього екрану).

Приклад когоспної ідальні :

2.1. Відкриваємо (створюємо) файл s_1.scx в генераторі екранів за допомогою команд:

```
clear
SET VIEW TO KOLG_ID.VUE
MODIFY SCREEN S_1.SCX (або - CREATE SCREEN S_1.SCX)
```

Спочатку працюємо з опцією Screen : Layout.... де задаємо загальні параметри вікна для екрану S_1.SCX. Результат роботи має такий вигляд на екрані генератора меню, як на мал.41.

Тексти програмних модулів з розділу Screen Code: [X] Setup... та [X] Cleanup & Proc... будуть приведені в п.2.5. цієї лабораторної роботи.

Опція Type.... де задано тип та параметри вікна, має такий вигляд, як на мал.42.

☐ Desktop

☒ Window

Name: s_1
 Title: Робота з накладними
 Footer: Tab - перехід між полями; Esc - Вихід

Size:

Screen Code:

Height: 19
Width: 70

☒ Setup...
☒ Cleanup & Procs...

Position:

READ Clauses:

Row:
Column:
☒ Center

☐ Activate...
☐ Valid...
☐ Deactivate...

☐ Show...
☐ When...

Environment:

☒ Add alias

Мал.41. Опція Screen Layout... для екрану S_1.SCX.

Type:

User

Attributes:

Border:

☒ Close
☒ Float
☒ Shadow
☒ Minimize

☐ None
☐ Single
☒ Double
☐ Panel
☐ System

Color Schemes:

Primary:

Popup:

Scheme 13

Scheme 13

Мал.42. Опція Type... для екрану S_1.SCX.

2.2. Після вибору типу та інших параметрів вікна маємо в генераторі екранів рамки та текстові фрагменти у потрібних місцях згідно ескізу. Генератор має тепер такий вигляд, як на мал.43.

Частинною задачею є розташування полів бази даних у потрібних місцях накладної. Мал.44. ілюструє створення нового екранного об'єкту - поля пов_накл бази d_nakl (з таким же форматом, як і у базі) - після виклику опції Field режиму Screen. При натисканні на кнопку «Get...» з'являється діалог (мал.45.), в якому легко вибрати потрібне поле бази даних. Якщо перейти на нього клавішею Tab та натиснути Enter (також легко вибрати іншу базу відкритого і, ще-файлу за допомогою кнопки Database:).

System File Edit Database Record Program Window Screen
S_1.SCX

R: 0 C: 43

Move

Робота з накладними

Накладна N				
від				
(внутрішній шифр)				
Постачальник				
кому	колгоспній ідальні			
Назва товару	Кількість	Од.вимір.	Ціна	Сума
Загальна сума накладної:				

Tab - перехід між полями; Esc - Вихід

Мал.43. Проміжковий вигляд генератору екранів.

Field:

() Say (t) Get () Edit

< Get... > d_nakl.nom_nakl

< Format... >

Range:

[] Upper... [] Lower...

[] When... [] Error... [] Scroll bar

[] Valid... [] Comment... [] Allow tabs

[] Message... [] Disabled [] Refresh

Мал.44. Створення поля nom_nakl бази d_nakl.

Choose a Field or Variable: (var)

d_nakl.nom_nakl

Database Fields:

NA_PROD	C
SH_NAKL	N
SH_OD_VYM	N
KILKIST	N
CINA	N

Database: RE_PROD

Memory Variables:

_ALIGNMENT	C
_BOX	L
_INDENT	N
_LMARGIN	N
_PADVANCE	C
_PAGEEND	N
_PBPAGE	N
_PCOLNO	N

o OK n

< Cancel >

Мал.45. Діалог для кнопки < Get... >.

Всі інші поля, необхідні для накладної, створюються аналогічно. Слід тільки пам'ятати, що при відкритті поля іншої бази дані при існуючих реляційних зв'язках автоматично виникне необхідний запис. Складніший вираз має поле для підрахування суми накладної, яке задається екраном, зображеним на мал.46.

Field:

(t) Say () Get () Edit

< Say... > str(sum_nal),8,2)

< Format... >

Range:

[] Upper... [] Lower...

[] When... [] Error... [] Scroll bar

[] Valid... [] Comment... [] Allow tabs

[] Message... [] Disabled [X] Refresh

Мал.46. Створення поля для підрахування суми накладної.

У цьому полі використано власну функцію `sum_na()`, текст якої приведено у розділі 2.5. Задіяна опція `[X] Refresh` забезпечує оновлення поля при будь-яких змінах інших полів екрану. В результаті проведених дій накладна з усіма необхідними полями має такий вигляд, як на мал.47.

System File Edit Database Record Program Window Screen
S_1.SCX

R: 5 C: 16 | Move |

Робота з накладними

Накладна N	1: nom_nakl
від	2: data_nakl
(внутрішній шифр	3: sh_n)
Постачальник	4: na_post.....
кому	колгоспний ідальні

Назва товару	Кількість	Од.вимір.	Ціна	Сума

Загальна сума накладної: 10: str(su

Tab - перехід між полями; Esc - Вихід

Мал.47. Це один проміжковий вигляд генератора екранів.

2.3. Найбільш складною задачею при генерації екрану є створення відповідних кнопок, що виконують задані функції. Так, для вибору постачальника під або над полем типу SAY (4: na_post....) вводиться невидима кнопка, процес розробки якої ілюструється наступним фрагментом екрану (мал.48.).

System File Edit Database Record Program Window Screen
S_1.SCX

R: 4 C: 16

Постачальн кому	Invisible Buttons #: 1	
	☐ Horizontal ☒ Vertical ☐ Terminating <Spacing...>	
	Variable:	o OK n
	< Choose... > p	< Cancel >
6: nnn....	Options:	
.....	☐ When... ☐ Comment... ☒ Valid... ☐ Disabled ☐ Message...	
.....	☐ Message...	
<Нова накла		

Мал.48. Створення невидимої кнопки для вибору постачальника.

В опції `Valid` розміщено наступний програмний код для активізації списку `ppr` типу `pprp` при натисканні на невидиму кнопку та тих дій, що після цього відбуваються:

```

c_a=select() && запам'ятовуємо номер роб.області у змінній c_a
pos=resno() && запам'ятовуємо номер запису у змінній pos
sele d_postac && відкриваємо базу d_postac
activate pprp rpp && активізація меню rpp, що спливає
sele d_nakl && переходимо у базу d_nakl
replace sh_postach with d_postac.sh_postach && записуємо в по-
&& ле sh_postach значення з поля sh_postach бази d_postac
select(c_a) && повертаємося в робочу область c_a
go pos && повертаємося на запис pos
show gets && оновлюємо всі поля типу GET екрана
return .t. && закінчення роботи функції

```


рядкової змінної чи поля та str (перетворення числової змінної або поля у рядок). Це зроблено задля того, щоб рядки списку нагадували реальну накладну. Вираз має такий вигляд (розберіть його!):

В результаті всіх проведених і описаних раніше дій екран накладної став таким, як показано на мал.50.

System File Edit Database Record Program Window Screen
S 1.SCX

R: 0 C: 0

Move

Робота з накладними

Накладна N

1: nom_nakl

від

2: data_nakl

(внутрішній шифр

3: sh_n)

Постачальник

.....

кому

колгоспній ідальні

Назва товару	Кількість	Од.вимір.	Ціна	Сума
6: pnp.....				
.....				
.....				
.....				

Загальна сума накладної:

10: str(su

Tab - перехід між полями; Esc - Вихід

Мал.50. Також проміжковий вигляд генератора екранів.

2.4. Розглянемо процес створення ще трьох типів кнопок. Вибір місяця для пошуку накладних забезпечує кнопка типу Popup, створення якої ілюструється екраном, зображенням на мал.51.

Текст програмного коду задіяної опції Valid дуже простий:

show gets

☒ List Popup
 ☐ Array Popup

січень

лютий

березень

квітень

травень

червень

липень

серпень

вересень

жовтень

Variable:

< Choose... > 11

Options:

☐ When...

☐ Comment...

☒ Valid...

☐ Disabled

☐ Message...

☐ 1st Element...

☐ # Elements...

Initial:

січень

о

OK

п

< Cancel >

Мал.51. Створення кнопки типу Popup.

Кнопки типу Push button використані для забезпечення пошуку накладних та введення нової накладної. Процес створення цих кнопок ілюструється наступним екраном (мал.52):

Програмний код з опції Valid має наступний вигляд:

```
DO CASE
CASE 111 = 1
DO SUBPR1  && виконати для режиму "Нова накладна"
CASE 111 = 2
DO SUBPR2  && виконати для режиму "Пошук"
ENDCASE
```

Push Button Prompts:

Нова накладна

Пошук

☐ Horizontal
 ☒ Vertical

☐ Terminating
 ☐ <Spacing...>

Variable:

< Choose... > 111

Options:

☐ When...

☐ Comment...

☒ Valid...

☐ Disabled

☐ Message...

о

OK

п

< Cancel >

Мал.52. Створення кнопки типу Push button.

Підпрограма SUBPR1 забезпечує введення нової накладної і знаходиться в розділі у вигляді окремого файлу SUBPR1.PRГ. Текст підпрограми такий:

```
sele c          && перейти в робочу область C
append blank    && додати запис
go bottom       && перейти в кінець бази
replace d_nakl.sh_nakl with resno() && записати в поле номер за-
                                     && пису
replace d_nakl.data_nakl with date() && записати в поле поточну
                                     && (системну) дату
f=d_nakl.sh_nakl && записати вибір накладної в змінну f
sele A          && перейти в робочу область A
set filter to   && знизити попередній фільтр на базі
append blank    && додати запис
go bottom       && перейти в кінець бази
replace re_prod.sh_nakl with f && записати в поле змінну f
set filter to   sh_nakl = f && накласти фільтр по полю sh_nakl
                                     && (показувати тільки ті записи бази, де sh_nakl = f)
show gets
```


sele C && перейти в робочу область C

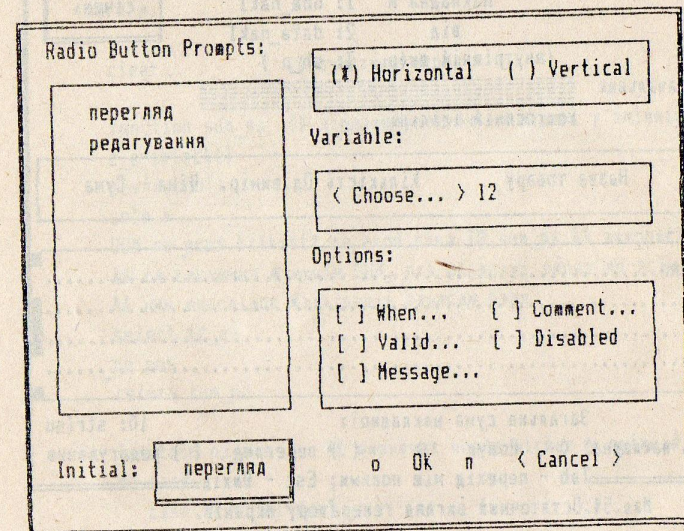
Підпрограма SUBPR2 забезпечує пошук потрібної накладної і також знаходиться в розділі у вигляді окремого файлу SUBPR2.PR6. Текст підпрограми такий:

```
pos=resno()    && запам'ятати поточний номер запису в змінній pos
go top        && перейти в початок бази
f1=str(na_mis(),5) && запам'ятати в рядковій змінній f1 номер
             && місяця, який передає власна функція na_mis()
set filter to str(month(d_nakl.data_nakl),5) = f1 && накласти
             && фільтр по полю data_nakl так, щоб рядок з 5-ти символів
             && функції month() від цього поля співпадав з рядком f1
if .not.bof() then && подвійний оператор IF для того, щоб яка-
                 && зівник запису змінив своє значення (для
                 && коректної роботи меню pp2)

go top
else
if .not.eof() then
go bottom
endif
endif
if .not.eof() then    && якщо не кінець файлу, тоді виконати
activate popup pp2 REST && активізація меню pp2, що спливає
set filter to        && зняти попередній фільтр
else                && інакше, тобто коли потрібні записи відсутні
set filter to        && зняти попередній фільтр
go pos              && перейти на запис pos
endif
f=d_nakl.sh_nakl    && занести в змінну f значення поля sh_nakl
sele A              && перейти в робочу область A
set filter to        && зняти попередній фільтр
set filter to sh_nakl = f && накласти фільтр по полю sh_nakl
sele C              && перейти в робочу область C
show gets
```

Текст функції na_mis(), що використана в останній підпрограмі, вміщено у вікні Cleanup (див.п.2.5.).

Остання кнопка екрану типу Radio Button дозволяє вибрати одну з двох альтернатив - переглядати чи редагувати накладну. Створення кнопки ілюструється екраном, зображеним на мал.53.



Мал.53. Створення кнопки типу Radio Button.

Змінна кнопки 12, як Ви пам'ятаєте, була використана раніше у п.2.3. для роботи списку типу List.

2.5. Виконання всіх попередніх дій призвело до наступного остаточного загального вигляду файлу S_1.SCX у генераторі екранів (мал.54.):

System File Edit Database Record Program Window Screen
S_1.SCX

R: 0 C: 0

Move

Робота з накладними

Накладна N	1: nom_nakl	січень																									
від	2: data_nakl																										
(внутрішній шифр	3: sh_n)																										
Постачальник																											
кому	кологоспін і даальні																										
<table border="1"> <thead> <tr> <th>Назва товару</th> <th>Кількість</th> <th>Од.вимір.</th> <th>Ціна</th> <th>Сума</th> </tr> </thead> <tbody> <tr> <td colspan="5">6: ppp.....</td> </tr> <tr> <td colspan="5">.....</td> </tr> <tr> <td colspan="5">.....</td> </tr> <tr> <td colspan="5">.....</td> </tr> </tbody> </table>			Назва товару	Кількість	Од.вимір.	Ціна	Сума	6: ppp.....																			
Назва товару	Кількість	Од.вимір.	Ціна	Сума																							
6: ppp.....																											
.....																											
.....																											
.....																											
Загальна сума накладної: 10: str(su <Нова накладна> < Пошук > () перегляд () редагування Tab - перехід між полями; Esc - Вихід																											

Мал.54.Остаточний вигляд генератору екранів.

Перед початком роботи екрану S_1 виконується програмний код опції Setup, який має наступний вигляд:

```
define popur ppp from 3,8 to 20,40 prompt fields na_post
  && визначення меню ppp, що спливає
  on selection popur ppp deactivate popur ppp && при виборі пот-
  && рібного пункту меню ppp останнє повинно закритися
  define popur pp2 from 3,8 to 20,50 prompt fields str(d_nakl.nom_nakl,
  5)+ " +DTOS(data_nakl)+ " +substr(d_postac.na_post, 1,20)
  && визначення меню pp2, що спливає
  on selection popur pp2 deactivate popur pp2 && при виборі пот-
  && рібного пункту меню pp2 останнє повинно закритися
  f=d_nakl.sh_nakl && занести в змінну f значення поля sh_nakl
  sele A && перейти в робочу область A
  set filter to sh_nakl = f && накласти фільтр по полю sh_nakl
```

sele C

&& перейти в робочу область C

Після закінчення роботи екрану S_1 виконується програмний код опції Cleanup, у якій також вищено всі функції користувача, що були використані у інших програмних фрагментах та згадані раніше в тексті лабораторної роботи. Текст з Cleanup має наступний вигляд:

```
clear && очистити екран

function sum_na && підраховує суму накладної у змінній sum_na
c_a=select()
pos=recno()
sele a
SUM re_prod.kalkist re_prod.cina TO sum_na && використовує-
&& ся системна функція SUM, яка обчислює вираз по всім доступ-
&& ним внаслідок фільтрації записам бази
select (c_a)
go pos
return sum_na

function na_mis && визначає номер місяця у змінній na_mis
do case
  case ll = "січень"
    na_mis=1
  case ll="лютий"
    na_mis=2
  case ll="березень"
    na_mis=3
  case ll="квітень"
    na_mis=4
  case ll="травень"
    na_mis=5
  case ll="червень"
    na_mis=6
  case ll="липень"
    na_mis=7
  case ll="серпень"
    na_mis=8
```



```
case 11="вересень"
  na_mis=9
case 11="жовтень"
  na_mis=10
case 11="листопад"
  na_mis=11
case 11="грудень"
  na_mis=12
endcase
return na_mis
```

Всі інші екрани, які потрібні для додатку до бази даних, створяться аналогічно, оскільки складаються з тих же самих екранних об'єктів, що були розглянуті вище. У згаданому вже розділі мережі EXAMPLE вміщено також файл EDIT_STR.SCX, який реалізує екран для другого ескізу по колгоспній ідальні з п.1.

3. Після усунення можливих недоліків та затвердження викладачем першого екрану для кожного з інших ескізів, зроблених в п.1, виконати наступні дії:

а) перейти у вікні View на потрібну базу, що є основною для даного екрану;

б) виконати п.2 для чергового екрану згідно відповідного ескізу.

4. Продемонструвати викладачу дім головного меню власного додатку до бази даних, яке може в потрібному місці викликати екрани s_1.spr, s_2.spr і т.і.

ВМІСТ ЗВІТУ

- Звіт з лабораторної роботи обов'язково повинен вмішувати:
- відповіді на питання по всім пунктам змісту роботи, де є згадка про звіт;
 - файли s_1.spr, s_2.spr і т.д. та файл m_main.spr, що їх використовує (в електронному вигляді).

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Що необхідно визначити для того, щоб намалювати ескіз потрібного для головного меню екрану?
2. Які функції використовуються для збереження номера робочої області та номера запису? Наведіть приклади.
3. Який формат має функція STR()? Наведіть приклади.
4. Який формат має функція SUBSTR()? Наведіть приклади.
5. Для чого використовуються опції Setup та Cleanup?
6. Що можна вставити в процедури Setup та Cleanup екрана?
7. Як забезпечується виклик 1.vue-файлу з екрану? Як запобігти подвійному завантаженню 1.vue-файлу?
8. Як забезпечити очищення екрану монітора після закінчення дії сгенерованого екрана?
9. Які додаткові програмні модулі використано при створенні екранів колгоспної ідальні? Навіщо це потрібно?
10. Що таке фільтр? Як його накласти та зняти?
11. Який формат має функція MONTH()? Наведіть приклади.
12. Як працює системна функція SUM? Який вона має формат?

ЛАБОРАТОРНА РОБОТА № 2.7

ТЕМА: СТВОРЕННЯ ЗВІТІВ ДЛЯ ДОДАТКУ ДО БАЗИ ДАНИХ ЗА ДОПОМОГОЮ ГЕНЕРАТОРА ЗВІТІВ (6 год.).

МЕТА РОБОТИ: ознайомлення з методиком створення 1.frx-файлів в системі FoxPro та основними можливостями генератора звітів на прикладі власного додатку до бази даних; набуття навичок розробки звітів та використання їх у програмних файлах.

СТИСЛІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Генератор звітів СУБД FoxPro відчутно простіший в користуванні, аніж вже знайомий Вам генератор екранів. Спільним в обох генераторах є концепція "візуального програмування", що побудована на понятті екранного об'єкту і яка в останній час дуже інтенсивно розвивається в сучасних програмних продуктах. Процеси

створення, корегування та маніпулювання екранними об'єктами в обох генераторах тому повністю аналогічні. Відмінності генератору звітів є значно вужчий набір об'єктів та неможливість створити (сгенерувати) програмний файл звіту. В результаті виникає необхідність використання у програмі спеціальних команд, які створюють у потрібний час звіт з *.frx-файлу.

При створенні або корегуванні звіту в головному меню з'являється нова опція Report, яка має наступну структуру:

Page Layout...	
Page Preview... ^I	
Data Grouping...	
Title/Summary...	
Variables...	
Box	^B
Field...	^F
Text	^T
Add Line	
Remove Line	^D
Bring to Front	^G
Send to Back	^J
Center	
Group	
Ungroup	
Quick Report...	

Коротенько про ці опції:

Page Layout... - задає параметри сторінки звіту;
 Page Preview... - перегляд поточного звіту;
 Data Grouping... - групування екранних об'єктів звіту;
 Title/Summary... - задає параметри заголовку та підписки звіту;
 Variables... - встановлює змінні;
 Box - рамка (екранний об'єкт);

Field... - поле (екранний об'єкт);
 Text - текст (екранний об'єкт);
 Add Line - додає рядок;
 Remove Line - знімає рядок;
 Bring to Front - змінює чергу об'єктів у групі по порядку їх занесення у групу;
 Send to Back - зворотня дія до попереднього пункту;
 Center - центрування об'єкта в рядку;
 Group - об'єднання об'єктів у групу;
 Ungroup - відміна об'єднання об'єктів у групу;
 Quick Report... - створення "швидкого звіту".

Детальну інформацію про всі ці опції Ви можете подивитися у відповідних розділах HELPU системи FoxPro.

ЗМІСТ РОБОТИ

1. Для свого варіанту додатку до бази даних розробити ескізи необхідних для головного меню звітів аналогічно до наведеного нижче прикладу колгоспної ідальні. Ескізи узгодити з викладачем та занотувати у звіт.

Приклад колгоспної ідальні ілюструє ескіз на мал.55.

РЕЕСТР ТОВАРІВ ІО КОЛГОСПНІЯ ІДАЛЬНІ

Назва товару	Дані накладної	Одиниця виміру	Кількість	Ціна	Сума
Картопля	23 03.02.1997	кілогр	1.000	3.00	3.00000
Буряк	23 03.02.1997	штука	2.000	5.00	10.00000
М'ясо свин.	15 03.03.1997	кілогр	130.000	5.20	676.00000
М'ясо волон.	15 03.03.1997	кілогр	68.000	3.40	231.20000
Морква	23 03.02.1997	кілогр	1.000	1.00	1.00000
Ковбаса московська	15 03.03.1997	кілогр	42.000	9.30	390.60000
Сметана	4 03.03.1997	літр	5.000	21.80	109.00000
Молоко	4 03.03.1997	літр	40.000	0.55	22.00000
Вершки	4 03.03.1997	літр	40.000	1.20	48.00000
Хліб чорний	87 28.02.1997	кілогр	74.000	0.68	50.32000
Сіль	11 02.03.1997	кілогр	1.000	1.00	1.00000
Яйця курячі	11 02.03.1997	штука	1.000	1.00	1.00000
21.05.97	Сторінка:	1	Сума по сторінці: 1543.12000		

Назва товару	Дані накладної	Одиниця виміру	Кількість	Ціна	Сума
Контейнери	23 03.02.1997	штука	1.000	1.00	1.00000
Ящики	15 03.03.1997	штука	10.000	3.50	35.00000
Бідони	4 03.03.1997	штука	3.000	8.80	26.40000
Апельсини бочками	11 02.03.1997	грам	1.000	1.00	1.00000
Молоко	66 11.11.1997	літр	10.000	0.50	5.00000
Сметана	66 11.11.1997	літр	58.000	1.00	58.00000
Ковбаса	77 11.11.1997	кілогр	55.000	10.00	550.00000

Загальна сума документа: 2219.52000

Завідуючий ідальнею: Бухгалтер:
 /Петренко О.О./ /Гуцалюк П.В./
 21.05.97 Сторінка: 2 Сума по сторінці: 676.40000

Мал.55. Ескіз звіту для реєстру товарів
 (відповідає опції меню: Вихідні документи ! Реєстр товарів).

2. Увійти в систему FoxPro, відкрити вікно View та запустити у вікні Command команду SET VIEW TO <ім'я>.VUE для відкриття свого *.VUE-файлу. У вікні View стати на ту базу даних, яка є головною для першого з звітів. Натиснути клавішу пропуск.

Приклад колгоспної ідальні - у вікні Command вводиться наступна послідовність команд:

```
clear
SET VIEW TO KOLG_ID.VUE
SELECT A
```

3. Відкрити новий файл в генераторі екранів : File ! New (вказати пропуском у стовпчику вибіра (!) Report). На екрані монітора буде відкрито спеціальне робоче вікно для створення нового звіту, а в головному меню праворуч з'явиться нова опція Report. Знайти в ньому і запустити режим Quick Report, у діалого, що з'явиться, нічого не міняти та натиснути Enter.

Приклад колгоспної ідальні : після виконання п.3 генератор звітів має такий вигляд, як на мал.56.

```
System File Edit Database Record Program Window Report
UNTITLED.FRX
R: 0 C: 0 || Move || Page Header ||
PgHead Na_prod Sh_nakl Sh_od_vym Kilcost Cna
PgHead
PgHead
Detail Na_prod sh_n sh kilcost cna
PgFoot
PgFoot
PgFoot DATE()
```

Мал.56. "Швидкий звіт" для файлу Re_prod.dbf.

4. Подивіться, як створюється та працює "швидкий звіт" для відкритої в системі FoxPro бази даних. Використовуючи стислі теоретичні відомості цієї та попередньої лабораторних робіт, відповідний HELP або провівши елементарні експерименти з генератором звітів, занотуйте в звіт відповіді на наступні питання:

- як працює опція Report ! Quick Report, якщо в поточний момент її активізації не відкрита жодна з баз даних?
- як записати у звіті текст та намалювати рамку?
- що можна вважати екранним об'єктом для звіту?
- як позначати об'єкти (групи об'єктів) та знімати позначення?
- як пересувати об'єкти (групи об'єктів) на екрані та змінювати їх розмір?
- як знищувати об'єкти (групи об'єктів) на екрані?
- як модифікувати екранні об'єкти?

5. Через File ! Save as... записати файл екрану під іменем R_QUICK.frx . Пересвідчитися в тому, що створити програмний файл R_QUICK звичайним шляхом Program ! Generate неможливо. Передивитися отриманий звіт двома способами:

- за допомогою опції Report ! Page Preview... або Ctrl+F (перехід на наступну сторінку - клавіша Tab);
- запустивши у командному вікні наступну групу команд:

REPORT FORM R_QUICK TO FILE fff.txt

clear

MODIFY COMMAND FFF.TXT

(в результаті звіт R_QUICK буде створено у файлі FFF.TXT).

Уважно передивитися текст створеного звіту та занотувати у звіт його структуру.

Приклад когоспної ідальні. Вигляд звіту по файлу R_QUICK.frx представлено на мал.57.

System File Edit Database Record Program Window Report
Preview

Na_prod	Sh_nakl	Sh_od_vym	Kilkist	Cina
Картопля	1	1	1.000	3.00
Буряк	1	2	2.000	5.00
М'ясо свин.	2	1	130.000	5.20
М'ясо волон.	2	1	68.000	3.40
Морква	1	1	1.000	1.00
Ковбаса московська	2	1	42.000	9.30
Сметана	3	3	5.000	21.80
Молоко	3	3	40.000	0.55
Вершки	3	3	40.000	1.20
Хліб чорний	4	1	74.000	0.68
Сіль	5	1	1.000	1.00
Яйця курячі	5	2	1.000	1.00
Контейнери	1	2	1.000	1.00
Ящики	2	2	10.000	3.50
Бідони	3	2	3.000	8.80
Апельсини бочками	5	4	1.000	1.00

o Done n < More > Column: 0

System File Edit Database Record Program Window Report

Preview				
Молоко	6	3	10.000	0.50
Сметана	6	3	58.000	1.00
Ковбаса	7	1	55.000	10.00

21.05.97

o Done n < More > Column: 0

Мал.57. Вигляд звіту по файлу R_QUICK.frx.

6. Проаналізувати результати, отримані внаслідок виконання попередніх пунктів змісту роботи. Занотувати у звіт відповіді на наступні питання:

- що відрізняє "швидкий звіт" від традиційних бухгалтерських звітів?
- які принципові можливості є у команді REPORT FORM?
- як реагує опція Report ! Quick Report на виклик в той час, коли не відкрита жодна з баз даних?
- що лишається на екрані монітора, якщо після команди REPORT FORM не давати команду CLEAR?
- чи створиться після чергового виклику команди REPORT FORM R_QUICK TO FILE fff.txt новий варіант файлу fff.txt?

(Очевидно, що даний звіт не задовільнить бухгалтера, оскільки у ньому не рахуються суми по кожній сторінці та загальна по звіту, а також присутні незрозумілі написи на іноземній мові та ще менш зрозумілі цифри накладних та одиниць виміру.)

7. Дати у вікні Command команду Clear. Користуючись набути-ми навичками та інформацією з стислих теоретичних відомостей, прикладом когоспної ідальні та підказками HELPU створити перший звіт згідно з першим ескізом п.1. Надати власне ім'я файлу звіту та продемонструвати дію цього звіту власного додатку викладачу.

Примітка: кожний звіт Вашого додатку до бази даних має за-довільняти наступним вимогам:

- Всі написи звіту зроблені українською мовою.
- Звіт знаходиться у вікні, для якого забезпечується весь стандартний сервіс роботи з вікнами.
- Забезпечується коректний виклик звіту з головного меню додатку та очищення екрану монітора після закінчення роботи.
- У звіті коректно підраховуються суми по кожній сторінці та загальна по звіту, відсутні всілякі цифри та автоматично ста-виться поточна дата звіту.

Приклад когоспної ідальні :

Відкриваємо (створимо) файл rees_tov.frx в генераторі ек-ранів за допомогою команд:

clear

SET VIEW TO KOLB_ID.VUE

MODIFY REPORT REES_TOV.FRX (a60 - CREATE REPORT REES_TOV.FRX)

Спочатку працюємо з опцією Page ! Layout..., де задаємо загальні параметри сторінки для звіту REPORT REES.TOV.FRX. Результат роботи має такий вигляд на екрані, як показано на мал.58.

Page Layout: < Options... >

Page length (rows): 20

Top margin (rows): 0

Bottom margin (rows): 0

Printer indent (columns): 0

Right margin column: 75

o OK n

< Cancel >

Environment:

< Save > < Restore > < Clear >

[] Printer Driver Setup...

Man.58. Onuia Page : Layout... 36ity REPORT REES TOV.FRX.

Зверніть увагу на те, що на екрані генератора звітів відсутні рядки для заголовку звіта і його закінчення. Для відкриття цієї можливості запускаємо опцію Report : Title/Summary... та маємо наступний вигляд вікна, що з'явилося (мал.59.):

Report Title:

☒ Title Band

☐ New Page

o OK n

Report Summary:

☒ Summary Band

☐ New Page

< Cancel >

Man.59. Onuir Report : Title/Summary...

Тепер додамо потрібну кількість рядків за допомогою опції Report : Add Line , намалюємо на екрані рамку та впишемо назви, потрібні для накладної. Результат має такий вигляд, як на мал.60.

```
System File Edit Database Record Program Window Report
```

REES TOV.FRX

R: 00: 01 Move Title
Title РЕЕСТР ТОВАРИВ ПО КОЛГОСПНІЯ ІДАЛЬНІ

Назва товару	Дані накладної	Одиниця виміру	Кіль- кість	Ціна	Сума
Сторінка: Сума по сторінці:					
Загальна сума документу:					
Завідуючий ідальнею:				Бухгалтер:	
/Петренко О.О./				/Гуцалюк П.В./	

Мал.60. Проміжковий вигляд генератора звітів.

Мал.60. Проміжковий вигляд генератору звітів.

Наступною задачею є введення необхідних полів з інформацією для накладної. Ці дані, що повторюються, заносяться в рядок Detail. Наприклад, для найбільш складної колонки "Дані накладної" вікно діалогу для введення параметрів поля має наступний вигляд (mal.61.):

Report Expression:
 < Expr... > str(d_nakl.nom_nakl)
 < Format... > Width: 16

[X] Style... [] Suppress...
 [] Calculate... [] Stretch Vertically
 [] Comment... [] Float as Band Stretches

o OK n < Cancel >

Мал.61. Діалог для введення параметрів поля.

Власне змінна, яка записана в опції < Expr... >. має такий же вигляд, як в екрані s_1.scx, а саме:

```
str(d_nakl.nom_nakl,5)+" "+substr(DTOS(d_nakl.data_nakl),7,2)+
"."+substr(DTOS(d_nakl.data_nakl),5,2)+
"."+substr(DTOS(d_nakl.data_nakl),1,4)
```

Для знайдення суми в Detail використано вираз, який ілюструється виглядом стандартного діалогу для введення складних змінних на мал.62.

Інші поля для рядку Detail створюються аналогічно. В результаті генератор звітів має такий вигляд, який зображено на мал.63.

Для суми по сторінці копіємо поле суми з Detail у потрібне місце накладної. Натискаємо на ньому Enter та пропуском активуємо опцію [X] Calculate... Остання має такий вигляд, який ілюструє мал.64.

Report <expr>:
re_prod.kilkist re_prod.cina

Field Names:	Database:	Variables:
NA_PROD C	RE_PROD	_ALIGNMENT C
SH_NAKL N		_BOX L
SH_OD_VYM N		_INDENT N
KILKIST N	< Verify >	_LMARGIN N
CINA N	o OK n	_PADVANCE C
	< Cancel >	_PAGE NO N
		_PB PAGE N
		_PCOL NO N

Мал.62. Створення поля для підрахування суми.

System File Edit Database Record Program Window Report
REES_TOV.FRX

R: 0 C: 0 Move Title
Title РЕЕС Т Р ТОВАРІВ ПО КОЛГОСПНІЯ ІДАЛЬНІ

Назва товару	Дані накладної	Одиниця виміру	Кількість	Ціна	Сума
Detail na_prod	str(d_nakl.nom_n_na_od_kilkist	cina	re_prod.k		
PgFoot	Сторінка:	Сума по сторінці:			
PgFoot					
Summary		Загальна сума документу:			
Summary					
Summary	Завідуючий Ідальнею:	Бухгалтер:			
Summary					
Summary	/Петренко О.О./	/Гуцалюк П.В./			

Мал.63. Ще один проміжковий вигляд генератору звітів.

Reset: End of Page

Calculate: () Nothing o OK n
() Count
(*) Sum < Cancel >
() Average
() Lowest
() Highest
() Std. Deviation
() Variance

Мал.64. Створення поля для визначення суми сторінки.

Аналогічна процедура для поля с загальною сумою по документу має результат, зображений на мал.65.

Reset:	End of Report	
Calculate:	☐ Nothing ☐ Count ☒ Sum ☐ Average ☐ Lowest ☐ Highest ☐ Std. Deviation ☐ Variance	OK < Cancel >

Мал.65. Створення поля для визначення суми документа.

Для поля з датою використовуємо систему функцій DATE(), а для поля з сторінкою – систему процедуру _PAG. Виконання всіх попередніх дій призвело до наступного остаточного загального вигляду файлу S_1.SCX у генераторі екранів (мал.66.):

```
System File Edit Database Record Program Window Report
      REES_TOV.FRX
R: 0 C: 0 Move Title
Title РЕЕС ТОВАРИВ ПО КОЛГОСПНІЙ ІДАЛЬНІ
PgHead
PgHead
PgHead Назва товару Дані Однаниця Кіль- Ціна Сума
PgHead накладної виміру кість
PgHead
Detail na_prod str(d_nakl.nom_n na_od_kilkist cina re_prod.k
PgFoot
PgFoot DATE() Сторінка: _PAG Сума по сторінці: re_prod.k
Summary
Summary Загальна сума документа: re_prod.k
Summary
Summary Завідуючий ідальнею: Бухгалтер:
Summary
Summary /Петренко О.О./ /Гуцалюк П.В./
```

Мал.66. Остаточний вигляд файлу S_1.SCX у генераторі екранів.

9. Після усунення можливих недоліків та затвердження викладачем першого звіту для кожного з інших ескізів, зроблених в п.1, виконати наступні дії:

а) перейти у вікні View на потрібну базу, що є основною для даного звіту;

б) виконати п.7 для чергового звіту згідно відповідного ескізу.

9. Продемонструвати викладачу дію головного меню власного додатку до бази даних, яке може в потрібному місці викликати звіт, створений в даній лабораторній роботі.

В М І С Т З В І Т У

Звіт з лабораторної роботи обов'язково повинен вмішувати:

- відповіді на питання по всім пунктам змісту роботи окрім пп.7,8,9;
- файли власних звітів з розширенням *.frx (в електронному вигляді).

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Якими принципами необхідно керуватися при побудові звітів додатку до бази даних?
2. Що необхідно визначити для того, щоб намалювати ескіз потрібного для головного меню звіту?
3. Які об'єкти для звітів використовує система FoxPro?
4. Як стандартні операції з об'єктами забезпечує генератор звітів FoxPro?
5. Що можна регулювати опцією Page Layout... ?
6. Як працює тип звіту Quick Report... ?
7. Як забезпечується виклик *.vue-файлу з звіту? Як запобігти подвійному завантаженню *.vue-файлу?
8. Як забезпечити очищення екрану монітора після закінчення дії звіту?
9. Які особливості бази даних використано при створенні звіту для колгоспної ідальні? Навіщо це потрібно?
10. Як відрізняється процедура (методика) знайдення суми для звіту від аналогічної для екрана? Яка процедура простіша?

ТЕМА: ДОСЛІДЖЕННЯ МЕТОДІВ ПОШУКУ ТА СОРТУВАННЯ ДАНИХ
(4 год.).

МЕТА РОБОТИ: ознайомлення з основними можливостями пошуку та сортування даних в системі FoxPro; набуття навичок відлагодження програмних функцій, потрібних для виконання запитів на мові SQL.

СТИСЛІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Найголовнішою функцією будь-якої СУБД вважається пошук та сортування даних. Особливо актуальною ця проблема стає для великих баз даних, в яких кількість записів перевершує 1000. В системі FoxPro існують досить багато різних можливостей для пошуку даних - як простих, так і більш складніших.

Найпростішим видом пошуку, який підтримує система FoxPro, вважається звичайний пошук послідовності символів, який працює у базі так само, як і в текстовому файлі. Викликається цей режим за допомогою опції Edit ! Find, яка активізує спеціальний діалог Find (Найти). Останній дозволяє вказати слово чи фразу для опцій Look For (Найти) або Replace With (Замінити на). Діалог Find, як і більшість аналогічних режимів пошуку в текстових процесорах, може ігнорувати прописні та заглавні літери, шукати по повному співпадінню комбінації символів, проводити пошук від початку файла до курсора або від курсора до кінця файла, тощо.

Але, звичайно, шукати в базі так, як і в текстовому файлі, зовсім недоцільно. Адже база даних має спеціальну структуру з полів та записів, яка по своїй суті дозволяє провести пошук чіткіше та конкретніше і, відповідно, набагато швидше. Тому в будь-якій СУБД існує режим пошуку по записам; в системі FoxPro це команда Locate, яку ще можна викликати з меню: Record/Locate... . Locate (Локалізувати) починає пошук від початку файла. Вказівник записів встановлюється на перший запис, що відповідає зазначеним умовам (For та/або While).

Діалог Locate дозволяє визначити таке:

[] Scope... - задається сфера дії, якщо вона відрізняється від ALL;

[] For... - передумова пошуку (якщо);
[] While... - післяумова пошуку (доки не).

На прикладі файла re_prod з бази даних колгоспної ідальні за допомогою наступної команди можна знайти такий запис, поле з назвою товару якого вміщує рядок 'лі':

```
LOCATE ALL FOR 'лі' re_prod.na_prod
```

Детальніше про особливості команди можна дізнатися у відповідному розділі HELPу системи FoxPro.

Одними з можливих варіантів глобального пошуку, які також виконуються по записам, можна вважати фільтрації та сортування бази даних. Фільтр накладається на базу даних за допомогою команди SET FILTER TO (умова фільтрації) та робить доступними ("видимими") для всіх інших команд FoxPro тільки ті записи, які задовольняють умові фільтрації. Треба пам'ятати, що фільтр накладається тільки на активну в даний момент базу даних, а вираз з умови фільтрації повинен відповідати синтаксису системи. Команда SET FILTER TO без умови фільтрації відміння попередній фільтр, і база даних знову стає повністю видимою в режимі Browse. Корисною функцією при фільтрації є FILTER(), яка показує той вираз, який вміщено в умову діючого фільтра.

На прикладі файла re_prod з бази даних колгоспної ідальні за допомогою наступної команди можна зробити доступними тільки такі записи, поле з назвою товару в яких вміщує рядок 'ма':

```
SET FILTER TO 'ма' re_prod.na_prod
```

Команда ? FILTER() у даному випадку покаже наступне:
'ма' re_prod.na_prod .

За допомогою режиму сортування або відповідної команди можна не тільки відібрати потрібні дані в окрему базу, але й розташувати їх у вибраному порядку. Полегає процес сортування діалог Sort, який знаходиться в опції Database ! Sort... головного меню. В цьому діалозі присутні такі опції:

Database Fields (List) - виведення полів активної бази даних;
< Move -> - кнопка для переміщення вибраних полів зі списку

Database Fields в список Set Order (встановити порядок);
 < Remove > - вибирає відмічені поля зі списку Set Order;
 () Ascending /по збільшенню/ або () Descending /по зменшенню/
 - вказує тип сортування для полів, що вибрані в списку Sort Order (порядок сортування);
 [] Ignore case - якщо вибрано, то при сортуванні не враховується регістр (верхній та нижній) літер;
 Sort Order (List) - виведення полів в порядку зменшення значимості при сортуванні.

Dialog Sort дозволяє також змінювати базу даних, сферу сортування (якщо вона відмінна від ALL викинутого фільтру), визначати умови типу For та/або While і вирази для полів (діалог Field Picker), а також зберігати відсортовані дані в базі з іншим ім'ям.

На прикладі бази даних когоспної ідальні з файлу re_prod за допомогою наступної команди можна створити таку відсортовану базу SORT.DBF, в якій присутні поля NA_PROD, KILKIST, CINA записів, де sh_nakl=1 (сортування проведено по зростанню поля CINA):

```
SORT TO C:\SORT.DBF ON RE_PROD.CINA FOR re_prod.sh_nakl = 1
FIELDS RE_PROD.NA_PROD, RE_PROD.KILKIST, RE_PROD.CINA
```

Метод пошуку інформації по записам (тобто послідовний пошук) стає неприйнятним для великих баз даних внаслідок своєї повільності. Тому в сучасних реляційних СУБД присутні інструменти швидкого пошуку на основі використання індексних файлів. До речі, в системі FoxPro 2.0 використана унікальна для операційної системи MS DOS технологія побудови індексних файлів RUSHMORE, яка має найкращі показники по швидкості серед всіх відомих СУБД.

Для створення нового індексного файлу (або підключення існуючого) в системі FoxPro 2.0 існує спеціальна кнопка < Add... > в опції Database ! Setup. Система підтримує два типи індексів - компаундний або одиночний. Компаундний індекс має одночасно декілька індексів для різних тегів, тобто індексних виразів (вони можуть вмішувати в собі будь-який дозволений синтаксисом FoxPro вираз з полів, функцій, системних змінних, тощо). До всіх цих тегів є доступ, якщо компаундний індексний файл відкрито. Одиночний індекс створюється тільки для одного виразу. Версія FoxPro 2.0 дозволяє створювати компактні (стислі) одиночні ін-

декси для зменшення їх розміру.

Процес створення нового індексного файлу забезпечується спеціальним діалоговим вікном, яке має наступні опції:

Database Fields:	виводить поля активної бази даних;
() Ascending	сортують поля в порядку зростання;
() Descending	сортують поля в порядку зменшення;
[] Unique	при виборі опції виводяться тільки унікальні поля (ті, що не повторюються);
< For... >	виводить діалог Expression Builder для створення фрази FOR;
Tag name	сиди вводиться ім'я тега;
< Move >	розміщує тег в списку індексного виразу;
< Remove/Edit >	видаляє тег з списку Index On;
Index On:	перераховує всі індексні теги або вирази;
< Expr... >	вводиться індексний вираз або викликається діалог Expression Builder для його побудови;
() IDX	створює індекс типу .IDX (одиночний);
[X] Compact	при виборі створює компактний індекс типу .IDX;
() CDX	створює компаундний індекс типу .CDX;
[X] Structural	при виборі створює структурований індекс .CDX;
< Save As... >	в це поле вводиться ім'я індексного файлу.

Найуніверсальнішим засобом роботи з будь-якою базою даних беззаперечно вважається спеціальна мова запитів SQL, яка є міжнародним стандартом в цій галузі для більшості сучасних СУБД. Система FoxPro 2.0 повністю підтримує стандартний формат мови SQL, який характерний для самих різних програмних та апаратних платформ. Докладніше про формат запиту можна подивитися в розділі SQL HELPу системи. Найпростіший варіант запиту до телефонного довідника має наступний вигляд:

```
* Приклад запиту - знаходження всіх телефонних абонентів з одного
* (вибраного) дому м.Вінниці
SELECT * ;
FROM Vinphone;
WHERE Street='ХМ.ВОССЕ' and House="000021";
INTO CURSOR query
BROWSE
```


Цей приклад необхідно запустити в системі FoxPro таким чином, щоб був доступним файл Vinphone.dbf. Вибір режиму BROWSE після виконання запиту дозволяє швидко подивитися його результати. Запит можна ускладнити, якщо необхідно знайти телефони тільки одного парадного:

```
SELECT * ;
FROM Vinphone;
WHERE Street='ФУН3Е' and val(House)=69;
and val(Appartamen))=1 and val(Appartamen)<=36;
INTO CURSOR query
BROWSE
```

Для більш детального роз'яснення досить складного формату мови SQL спробуйте розібратися в наведених нижче п'яти прикладах з конкретними запитами до бази даних когоспної ідальні. Результати цих запитів можна отримати, якщо запустити їх на виконання при вже звантаженому файлі KOLG_ID.VUE (вже відомою командою SET VIEW TO KOLG_ID.VUE).

Приклад # 1 - Вибір максимальної, мінімальної і середньої цін з бази даних реєстра продуктів RE_PROD

```
SELECT MAX(cina) AS max, MIN(cina) AS min, AVG(cina) AS avg;
FROM RE_PROD;
INTO CURSOR q1
BROWSE FIELDS ;
max:h='max ціна':p='999.99',;
min:h='min ціна':p='999.99',;
avg:h='середня ціна':p='999.99',;
TITLE 'Загальні відомості про ціни на продукти в ідальні'
SELECT MAX(cina) AS max, MIN(cina) AS min, AVG(cina) AS avg;
FROM RE_PROD;
TO FILE prim1.txt
```

Приклад # 2 - Вибір всіх продуктів, що відносяться до накладної 2 з бази даних реєстра продуктів RE_PROD

```
SELECT * ;
FROM RE_PROD;
WHERE SH_NAKL=2;
```

```
INTO CURSOR q2
BROWSE FIELDS ;
NA_PROD :h='Назва продукту':20,;
D_NAKL.DATA_NAKL :h='Дата накладної',;
SH_OD_VYM :h='Одиниці виміру':10,;
KILKIST :h='Кількість':P='999,999.99',;
CINA :h='Ціна':P='999.99',;
TITLE 'Результат запроса в БД'
SELECT * ;
FROM RE_PROD;
WHERE SH_NAKL=2;
TO FILE prim2.txt
```

Приклад # 3 - Вибір всіх малоцінних продуктів з бази даних реєстра продуктів RE_PROD

```
SELECT * ;
FROM re_prod;
WHERE kilkist#cina<=8;
INTO CURSOR q3
BROWSE ;
FIELDS ;
NA_PROD :h='Назва продукту':20,;
SH_NAKL :h='Дата накладної',;
SH_OD_VYM :h='Одиниці виміру':14,;
KILKIST :h='Кількість':P='999,999.99',;
CINA :h='Ціна':P='999.99',;
TITLE 'Результат запроса в БД'
SELECT * ;
FROM re_prod;
WHERE kilkist#cina<=8;
TO FILE prim3.txt
```

Приклад # 4 - Вибір всіх накладних, дата яких збігається з точним місяцем + додатні кількість років до номера накладної

```
SELECT nom_nakl, DAY(data_nakl) as data_nakl,;
year(date())-year(data_nakl) as let,;
10000+nom_nakl as n_nom;
FROM d_nakl;
WHERE MONTH(data_nakl)=MONTH(DATE());
```



```

      INTO CURSOR q4
      BROWSE FIELDS ;
      nom_nakl :h='Номер накладної';3,;
      data_nakl :h='День створення';,;
      let :h='Пролетжала років!!!';,;
      n1=n_nom*let:h='Новий номер':P='999999';
      TITLE 'Результат запроса в БД '+DTOC(DATE())
      SELECT nom_nakl, DAY(data_nakl) as data_nakl,;
             year(date())-year(data_nakl) as let,;
             10000+nom_nakl as n_nom;
      FROM d_nakl;
      WHERE MONTH(data_nakl)=MONTH(DATE());
      TO FILE prim4.txt
  
```

† Приклад № 5 - Вибір номерів всіх накладних, кількості позицій
 † у них та значень загальних сум по накладній

```

      SELECT sh_nakl, COUNT(*) AS kol, SUM(kilkist*cina) AS sum_na;
      FROM re_prod;
      GROUP BY sh_nakl;
      INTO CURSOR q5
      BROWSE FIELDS ;
      sh_nakl:h='Шифр накл.';
      kol :h='Число позицій';
      sum_na :h='Сумма накладної';
      TITLE 'Результат запроса в БД '
      SELECT podr, COUNT(*) AS kol, SUM(zar) AS sum_zar;
      FROM kadry;
      GROUP BY podr;
      TO FILE prim5.txt
  
```

З М І С Т Р О Б О Т И

1. Зайти в систему FoxPro. Відкрити на диску Z:\FOXPRO\ файл VINPHONE.DBF командою USE Z:\FOXPRO\VINPHONE.DBF (або через головне меню: File!Open, потім вказати диск Z:\, вибрати каталог FOXPRO, а вже в ньому знайти файл VINPHONE.DBF).

2. Провести пошук рядкової змінної за допомогою звичайного текстового пошуку. Для цього увійти в режим BROWSE, далі перейти

в головному меню в режим пошуку - опції Edit ! Find, потім вказати для вікна Look For рядкову змінну (наприклад, "МЕЛЬНИК") і натиснути кнопку Find. За допомогою годинника з секундною стрілкою визначити час знаходження потрібної інформації та занотувати його у звіт разом з тою ознакою, яка показує закінчення процесу пошуку.

3. Провести такий самий пошук, як і у попередньому пункті, але за допомогою команди LOCATE. Наприклад (для рядка "МЕЛЬНИК"):

```
LOCATE ALL FOR 'МЕЛЬНИК' $ vinphone.name
```

Занотувати у звіт час виконання пошуку по записам та ознаку його закінчення.

Примітка: для того, щоб не гратися з годинником, можна використати вбудовану функцію FoxPro. Для цього треба створити невеличкий програмний файл з таким змістом:

```

t1=SECONDS()
LOCATE ALL FOR 'МЕЛЬНИК' $ vinphone.name
t2=SECONDS()
? " Час виконання в сек:",t2-t1
  
```

Спробуйте записати цей файл під назвою Time.prg (File ! New, потім вибрати Program; у вікні, що відкрилося, набрати текст програми; натиснути Ctrl+W; у вікні, що відкрилося, набрати ім'я програмного файлу). Порівняйте один та другий час, а висновки щодо цього занотуйте у звіт.

4. Накласти фільтр на базу з телефонним довідником за допомогою наступної команди:

```
SET FILTER TO 'МЕЛЬНИК' $ vinphone.name
```

(інший спосіб - кнопка [X] Filter... в опції Database!Setup...: в діалоговому вікні, що з'явиться, вказати вираз для фільтрації бази). Зафіксувати на основі модифікованої програми Time.prg час виконання цієї команди. Спробуйте тепер подивитися на відфільтровану базу (BROWSE LAST) та занотуйте свої враження щодо комфортності роботи з нею у звіт (не забудьте, що відміна фільтру - SET FILTER TO). Виконати такі самі дії для власної бази даних аналогічно прикладу, наведеному в стислих теоретичних відомостях. Результат фільтрації показати викладачу та занотувати у звіт.

5. Відсортувати базу з телефонним довідником за допомогою наступної команди:

СОРТ TO C:\SORT.DBF ON vinphone.phone FOR 'МЕЛЬНИК' \$vinphone.name
Зафіксувати на основі модифікованої програми Time.prg час вико-
нання цієї команди. Перегляньте тепер відсортовану базу
SORT.DBF, порівняйте свої враження щодо роботи з відфільтрованою
та відсортованою базом і занотуйте їх у звіт. Виконати такі самі
дії для власної бази даних аналогічно прикладу, наведеному в
стислих теоретичних відомостях. Результат сортування показати
викладачу та занотувати у звіт.

6. Створити на диску C:\ компаундний індексний файл по 3-м
полям: name, phone, street. Для цього в опції Database ;
Setup... вибрати кнопку Add..., потім вибрати диск c:\ (для то-
го, щоб на нього записався новий індексний файл) та вказати ре-
жим New. Із вікна Database Fields поля переносяться у вікно
Index On натисканням пропуску, а після нього - Enter. Загальний
вигляд діалогу для створення індексного файлу на ім'я IND_1.CDX
має бути таким (перед натисканням кнопки o OK n):

System File Edit Database Record Program Window

Database Fields:		Options	Index On:
NAME	C	(.) Ascending	NAME
PHONE	C	() Descending	PHONE
STREET	C	[] Unique	STREET
HOUSE	C	<For...>	
APPARTAMEN	C	Tag Name NAME	
		< Move >	
		<Remove/Edit >	
Index Expression <Expr...>		Output	
		() IDX [] Compact	
		(.) CDX [] Structural	
		<Save As...>	
		C:\IND_1.CDX	
			o OK n
			< Cancel >

Примітка: якщо індексний файл вже створено, то можна його
просто відкрити за допомогою команди: SET INDEX TO C:\IND_1.CDX;
інший шлях відкриття існуючого індексного файлу через меню:
Database!Setup...!Add..., тепер вибрати потрібний індексний файл
IND_1.CDX .

7. Для створеного (відкритого) індексного файлу активізува-
ти потрібний терг (NAME) для проведення пошуку. Це можна досягти
командою SET ORDER TO TAG NAME OF C:\IND_1.CDX або через головне
меню: Database ; Setup..., далі кнопка Set Order, а потім вказа-
ти поле NAME. Ознакою правильності проведених дій є значення
NAME в опції Index expr.

8. Провести пошук рядку 'МЕЛЬНИК' за допомогою команди
seek 'МЕЛЬНИК' . Перейти у вікно Browse та пересвідчитися, що
пошук проведено правильно. Зафіксувати на основі модифікованої
програми Time.prg час виконання цієї команди і занотувати свої
висновки у звіт.

9. Виконати дії пп.7,8 для двох інших тергів: phone та
street (наприклад, для номера телефону: seek '322130'). Звичай-
но, що інформацію для пошуку Ви можете вибрати будь-якою. Свої
враження від змін у вікнах Browse та загальних можливостей робо-
ти з індексними файлами занотуйте у звіт.

10. Для власної бази даних створити індексні файли та проде-
монструвати викладачу можливості пошуку на їх основі. Занотуйте
у звіт послідовність команд створення індексів для своєї бази.

11. На основі прикладів 1,2,3,4,5 стислих теоретичних відо-
мостей аналогічно скласти 5 своїх запитів до власної бази даних.
Створені запити перевірити, показати викладачу та записати їх
тексти разом з результатами запитів у звіт.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Дайте визначення понять пошуку та сортування даних. Наведіть конкретні приклади.
2. Що таке текстовий пошук? Які опції текстового пошуку підтри-
мує система FoxPro 2.0?
3. Як діє пошук по записам?
4. Які можливості має диалог Locate?
5. Що таке фільтрація? Які можливості має система FoxPro 2.0

Загальні вимоги до виконання та оформлення лабораторних робіт	2
1. Лабораторна робота 2.1: вивчення особливостей інтерфейсу інтерпретатора FoxPro 2.0	3
2. Лабораторна робота 2.2: створення взаємопов'язаної структури бази даних у вигляді *.DBF-файлів та заповнення її даними	7
3. Лабораторна робота 2.3: створення реляційних зв'язків між *.DBF-файлами бази даних та дослідження основних команд навігації	14
4. Лабораторна робота 2.4: створення головного меню додатку до бази даних за допомогою генератора меню	24
5. Лабораторна робота 2.5: принципи побудови генератора екранів СУБД FoxPro та методика його використання	37
6. Лабораторна робота 2.6: створення складних екранів для додатку до бази даних за допомогою генератора екранів	52
7. Лабораторна робота 2.7: створення звітів для додатку до бази даних за допомогою генератора звітів	73
8. Лабораторна робота 2.8: дослідження методів пошуку та сортування даних	86
9. Індивідуальні варіанти наскрізних об'єктів згортково- лого спрямування	96

Навчальне видання

Методичні вказівки до виконання лабораторних
дисципліни "Програмне забезпечення ЕОМ", частини
та СУБД (FoxPro 2.0) для студентів спеціаль-
70.50.106.

Склали: Олег Володимирович Бісікало
Евген Анатолійович Паламарчук
Сергій Феодосійович Роботко
Дмитро Володимирович Бісікало