МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

ОРГАНІЗАЦІЯ БАЗ ДАНИХ І ЗНАНЬ

методичні вказівки до виконання курсового проекту

для студентів напрямку підготовки
6.050101 “Комп’ютерні науки”
усіх форм навчання

2014

Організація баз даних і знань. Методичні вказівки до виконання курсового проекту для студентів напрямку підготовки 6.050101 “Комп’ютерні науки” усіх форм навчання /укл. С.К.Корнієнко, – Запоріжжя: ЗНТУ, 2014. – 34 с.

Укладач: С.К.Корнієнко, доцент, к.т.н.

Рецензент: В.П.Пінчук, доцент, к. ф.-м.н.

Відповідальний
за випуск: В.І.Дубровін, професор, к.т.н.

Затверджено
на засіданні кафедри
"Програмних засобів"

Протокол № 2
від 16.10.2013 р.

ЗМІСТ

1 Загальні положення 4
1.1 Мета курсового проектування 4
1.2 Тематика курсових проектів 4
2 Структура курсового проекту 5
2.1 Склад пояснювальної записки 5
2.2 Склад основної частини пояснювальної записки 5
2.3 Додатки 5
3 Методичні рекомендації до змісту основних розділів пояснювальної записки 6
3.1 Вибір і обґрунтування структури системи, що проектується 6
3.2 Концептуальне проектування 6
3.2.1 Технологія аналізу предметної області 6
3.2.2 Виявлення інформаційних об’єктів та зв’язків між ними 7
3.2.3 Побудова інформаційної структури 9
3.3 Логічне проектування системи 9
3.3.1. Вибір СКБД 10
3.3.2 Відображення концептуальної схеми на логічну схему 12
3.3.3 Проектування додатків 15
3.4 Проектування інтерфейсу користувача 17
3.5 Робота користувача з системою 20
Рекомендована література 22
Додаток А Приклад складення реферату 24
Додаток Б Приклад змісту пояснювальної записки 25
Додаток В Приклад оформлення технічного завдання 26
Додаток Д Приклади форм 30
Додаток Ж Словник основних термінів 33

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Мета курсового проектування

Мета виконання курсового проекту - набути практичних навичок у галузі проектування інформаційного забезпечення автоматизованих систем різного призначення.

В процесі курсового проектування студенти повинні:

– узагальнити, закріпити та поглибити знання, отримані під час вивчення дисципліни, використати їх для обґрунтованого прийняття проектних рішень;

– ознайомитись з методикою та практично закріпити навички комплексної розробки інформаційних систем в цілому та їх базових компонентів: програмного, інформаційного, лінгвістичного та математичного забезпечень.

– придбати досвід в оформленні відповідної проектної документації та складанні пояснювальних та розрахункових записок.

1.2 Тематика курсових проектів

Тематика курсових проектів повинна бути актуальною, відповідати сучасному стану розвитку науки та техніки, враховувати реальні потреби виробництва.

Приклади тем курсових проектів

- Автоматизована система обліку контрактів.
- Автоматизована система з продажу автомобілів.
- Автоматизована система складського обліку.
- Інформаційно-пошукова система "Бібліотека".
- Автоматизована система комерційної діяльності;
- Автоматизована система розрахунку заробітної плати;
- Автоматизація комерційної діяльності комп’ютерного центру.
- Автоматизація обліку господарської діяльності будівельно-ремонтної фірми.
- Розробка АРМу робітника планово-економічної служби.

За узгодженням із викладачем допускається самостійний вибір теми студентом.

2 СТРУКТУРА КУРСОВОГО ПРОЕКТУ

Курсовий проект складається з пояснювальної записки, та працездатної програмної системи на відповідному носії інформації.

2.1 Склад пояснювальної записки

Пояснювальна записка (ПЗ) повинна містити:

– титульний лист;
– завдання на проект;
– реферат (зразок наведений у Додатку А);
– зміст;
– перелік скорочень;
– основний текст;
– додатки.

2.2 Склад основної частини пояснювальної записки

Основний текст повинен складатися з таких розділів:
– вибір та обґрунтування проектних рішень;
– концептуальне моделювання предметної області;
– логічне проектування системи;
– розробка інтерфейсу користувача;
– методика роботи користувача з системою;
– висновки;
– перелік посилань.

Обсяг основної частини пояснювальної записки (за винятком додатків) - 30-40 сторінок друкарського тексту. Приклад змісту ПЗ наведений у додатку Б.

2.3 Додатки

Додатки повинні містити :
– приклади звітів;
– тексти програм;

💣 <u>Пояснювальна записка оформлюється відповідно з вимогами діючих стандартів</u>

3 МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ЗМІСТУ ОСНОВНИХ РОЗДІЛІВ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Всі розділи пояснювальної записки повинні містити необхідні теоретичні відомості з обов'язковим посиланням на відповідні джерела інформації.

3.1 Вибір і обґрунтування структури системи, що проектується

В цьому розділі потрібно на підставі аналізу технічного завдання на систему (оформленого або окремим підрозділом цього ж розділу, або додатком до ПЗ):

– структурувати інформацію на вході та виході системи;
– визначити форму представлення цієї інформації;
– структурувати інформацію за терміном її зберігання (оперативна, довідкова, архівна);
– чітко визначитися із задачами, що випливають із призначення системи;
– на підставі визначених задач розробити структуру програмного забезпечення;
– сформулювати вимоги до технічного забезпечення системи.

Якщо технічне завдання оформлюється окремим документом у додатку, то його потрібно виконувати згідно з прикладом, наведеним у Додатку В даних методичних вказівок.

3.2 Концептуальне проектування

3.2.1 Технологія аналізу предметної області

Аналіз предметної області (ПО) являє собою перший етап проектування БД будь-якого типу й закінчується побудовою інформаційної структури (концептуальної моделі). На цьому етапі аналізуються потреби користувачів, вибираються інформаційні об'єкти та їх характеристики, структурується предметна область. Аналіз ПО є загальним етапом, незалежним від програмного та технічного середовища, в яких буде реалізовуватися БД.

Аналіз предметної області доцільно розбити на три фази:

– аналіз концептуальних вимог та інформаційних потреб;
– виявлення інформаційних об`єктів і зв`язків між ними;
– побудова концептуальної моделі ПО та проектування концептуальної схеми БД .

На етапі аналізу концептуальних вимог та інформаційних потреб необхідно вирішити такі завдання:

– аналіз вимог користувачів до БД (концептуальних вимог);
– виявлення існуючих задач по обробці інформації (перспективних додатків);
– документування результатів аналізу.

Вимоги користувачів до БД, що розробляється, являють собою список запитів із зазначенням їх інтенсивності та обсягів даних.

На цьому етапі також з`ясовуються вимоги до виводу, поновленню та коректуванню інформації. Вимоги користувача уточнюються та доповнюються при аналізі існуючих та перспективних направлень.

3.2.2 Виявлення інформаційних об’єктів та зв’язків між ними.

Друга фаза аналізу предметної області складається з вибору інформаційних об’єктів, завдання необхідних характеристик для кожного об`єкта, визначення обмежень, які накладаються на інформаційні об`єкти, виявлення зв`язків між об`єктами та їх типів.

Під час вибору інформаційних об`єктів бажано відповісти на ряд запитань:

а) На які класи можна розподілити дані, що підлягають зберіганню у БД?

б) Яке ім`я можна надати кожному класу даних?

в) Які найбільш цікаві характеристики (з точки зору користувача) кожного класу даних можна виділити?

г) Які імена можна присвоїти вибраним наборам характеристик?

Виділення інформаційних об`єктів - процес інтерактивний. Він здійснюється на підставі аналізу інформаційних потоків та інтерв`ювання споживачів. Характеристики інформаційних об`єктів визначаються тими ж методами.

Далі виділяються зв`язки між інформаційними об`єктами. Під час цього процесу потрібно спромогтися відповісти на такі питання:

а) Які типи зв`язків між інформаційними об`єктами?
б) Яке ім`я можна присвоїти кожному типу зв`язків?
в) Які можливі типи зв`язків можна використовувати?
г) Чи мають сенс які-небудь комбінації типів зв`язків?

При необхідності потрібно завдати обмеження на об`єкти, їх характеристики та зв`язки, даючи відповіді на такі запитання:

а) Яка область значень для числових характеристик?
б) Які функціональні залежності між характеристиками одного інформаційного об`єкта?

Під обмеженням цілісності (внутрішньої єдності), як правило, розуміють логічні обмеження, які пред`являються до даних. Обмеження цілісності - це така властивість, яку ми задаємо для будь-якого інформаційного об`єкта або його характеристик і яка повинна зберігатися для кожного стану об`єкта.

Наприклад, твердження, що оклад професора ВНЗ більш ніж оклад доцента, може розглядатися як обмеження цілісності лише у тому випадку, якщо воно слушне в будь-який момент і не залежить від процедури підвищення заробітної плати, або зміни, наприклад, тарифної сітки.

В загальному випадку розглядаються три види обмежень: внутрішні, явні та змішані. Внутрішні обмеження в основному пов’язані з моделлю даних, яка підтримується СКБД.

Між різними інформаційними об`єктами, а також між інформаційним об`єктом та його характеристиками виникають певні асоціації, які називаються зв`язками. При проектуванні БД прийнято розглядати взаємозв`язки трьох типів: “один до одного”, “один до багатьох”, “багато до багатьох”.

3.2.3 Побудова інформаційної структури

Заключна фаза аналізу ПО - проектування її інформаційної структури (концептуальної схеми, або концептуальної моделі). Концептуальна модель використовується для структурування ПО з урахуванням не тільки інформаційних інтересів користувачів системи, але й інформаційних потреб самої предметної області.

В рамках кожної БД концептуальні вимоги узагальнюються в концептуальну модель, яка виражається абстрактними засобами, що дозволяє побачити весь інформаційний зміст предметної області.

При концептуальному проектуванні як основна модель використовується модель типу “сутність - зв`язок” (ER-модель). На мові ER-моделі інформаційний об`єкт називають сутністю.

Сутність слід визначати поіменованими характеристиками, які називаються атрибутами. Найменування повинно бути унікальним для кожного екземпляру сутності, хоча воно може повторюватись для різних типів сутностей.

В кожному наборі атрибутів, які характеризують сутність, необхідно вибрати ключеві атрибути, тобто атрибути, які однозначно ідентифікують кожний екземпляр сутності.

Після визначення сутностей та атрибутів виявляються залежності між двома і більше сутностями, сутностями та атрибутами, зв`язки атрибутів між собою. Кожний тип зв’язку іменується.

В цьому розділі на підставі аналізу предметної області системи необхідно навести:

- опис всіх сутностей зі своїми атрибутами у вигляді таблиць;
- перелік зв’язків між сутностями з указівкою типу зв’язку;
- перелік можливих обмежень на значення окремих атрибутів;
- ERдіаграму концептуальної моделі предметної області.

3.3 Логічне проектування системи

Цей розділ є фактично основним по розробці системи. В ньому повинні бути наступні підрозділи:

- вибір і обгрунтування інструментарію проектування;
- розробка схеми бази даних;
- нормалізація схеми бази даних;
- проектування запитів;

– проектування макросів;
– проектування додатків на мові Visual Basic;
– проектування звітів.

3.3.1. Вибір СКБД

Спочатку необхідно дати детальний огляд існуючих СКБД, інших засобів проектування сучасних БД. Вибір інструментарію проектування необхідно виконувати як за функціонально-технічними характеристиками засобів, так і за економічними покажчиками: вартістю та іншими.

При виборі СКБД, яка реалізує конкретну БД, виділяють два підходи до її оцінювання. Перший підхід пов`язаний з поглядом користувача на конкретну систему, інший пов`язаний із продуктивністю системи. Розглянемо параметри, що можуть характеризувати СКБД при її виборі.

Дані параметри не претендують на вичерпну повноту, проте дають достатньо повну картину про можливості СКБД:

1. Загальні характеристики:

– тип комп`ютера;
– максимальне число записів у файлі;
– максимальна ємність файлу;
– максимальне число символів на запис;
– максимальне число індексів на файл;
– максимальне число таблиць та операцій об`єднання;
– максимальне число файлів, які доступні для однієї команди;
– максимальне число одночасно відкритих файлів;
– максимальне число файлів в БД;
– максимальне число записів у БД;
– максимальне число змінних;
– максимальне число символів у одному полі;
– імпорт - експорт даних;
– тип внутрішньої моделі даних;
– фірма - виробник;
– версія.

2. Керування файлами та пошук:

- тип зв`язку;
- модифікація декількох файлів;
- двоспрямоване поєднання таблиць;
- мова маніпулювання даними;
- тип пошуку.

3. Засоби підтримки додатків:

- каталог даних;
- генератор додатків, процедурна мова та підпрограми;
- макроси;
- налогоджувач;
- система підтримки виконання;
- шифрування програм і даних;
- розмежування доступу;
- графіка;
- статистика.

4. Ввід та підтримка цілісності:

- керування за допомогою команд;
- перевірка цілісності за таблицею;
- перевірка унікальності ключа;
- перевірка за датою;
- незалежність даних.

5. Звіти:

- звіти за декількома файлами;
- зберігання форматів звітів;
- видача звіту на екран;
- поля, які обчислюються;
- перевизначення формату даних;
- генератор звітів;
- підсумкові поля;
- максимальна ширина звіту;

6. Операційне середовище:

– тип операційної системи;
– ємність потребної оперативної пам`яті;
– мова системи.

7. Додаткові відомості:

– наявність варіанту мережі;
– вартість.

3.3.2 Відображення концептуальної схеми на логічну схему

Для відображення інформаційної схеми, отриманої на етапі концептуального проектування, на логічну схему БД необхідно знати:

– розмір БД (число сутностей та атрибутів);
– частоту звернень до БД (число виконання додатків за одиницю часу);
– правила побудови логічних схем БД для існуючої у розпорядженні проектувальника СКБД;
– перелік додатків із вказівками на використані дані;
– конфігурацію, швидкодію та ємність пам`яті, яка надається у розпорядження проектувальника.

При відображенні інформаційної схеми кожний прямокутник схеми для ER-діаграми предметної області відображається у таблицю, яка являє собою одне відношення. При цьому потрібно враховувати обмеження на розмір таблиць, котрі накладає конкретна СКБД.

Припустимо, що на етапі концептуального проектування було складено дворівневе меню.

I рівень
1. Споживач
2. Виріб
3. Склад
4. Виробник
5. Заявка

II рівень

1. Споживач
1.1. Адреса
1.2. Телефон
1.3. Профіль магазину

2. Заявка
2.1. № заяви
2.2. Дата
2.3. Назва виробу

3. Склад
3.1. № складу
3.2. Адреса
3.3. Телефон
3.4. Профіль

4. Виробник
4.1. Назва
4.2. Адреса
4.3. Телефон
4.4. Профіль

5. Виріб
5.1. Назва
5.2. Вартість
5.3. Артикул
5.4. Шифр
5.5. Матеріал

Інформаційні об`єкти (сутності) будуть відповідати назвам першого рівня меню.

Зробимо відображення концептуальної схеми на реляційну схему та зробимо оцінку потрібної ємності пам`яті для зберігання БД. Як результат отримаємо наступні відношення:

Таблиця 3.1 - Відношення СПОЖИВАЧ

назва	адреса	телефон	профіль магазину
20	30	7	23

На один запис необхідно 20+30+7+23=80 байт. Всього споживачів 200. Таким чином, для зберігання цього відношення потрібно 80×200=16000 байт.

Таблиця 3.2 - Відношення ЗАЯВКА

№ заявки	дата	назва виробу
5	6	30

На один запис необхідно 41 байт. Всього заявок 1000, разом 41×1000=41000 байт.

Таблиця 3.3 - Відношення СКЛАД

№ складу	адреса	телефон	профіль
5	18	7	30

На один запис необхідно 60 байт. Всього складів 40, разом 60×40=2400 байт.

Таблиця 3.4 - Відношення ВИРОБНИК

назва	адреса	телефон	профіль
15	18	7	30

На один запис необхідно 70 байт. Всього виробників 15, разом 70×15=1050 байт

Таблиця 3.5 - Відношення ВИРОБ

назва	вартість	артикул	шифр	матеріал
30	5	5	5	15

На один запис необхідно 30+5+5+5+15=60 байт. Всього виробів 400, разом 60×400=24000 байтів.

Отже, в першому наближенні, на зберігання БД необхідно 16000+2400+1050+41000=84450 байт.

В цьому підрозділі необхідно в табличній формі навести схеми всіх відношень БД із переліком атрибутів, їх типів і розмірів. Для кожного відношення також необхідно підрахувати приблизний обсяг у байтах і ці розрахунки звести до окремої таблиці.

Після попереднього формування схеми бази даних її необхідно нормалізувати, тобто зробити декомпозицію попередніх відношень БД на більш прості відношення. Нормалізація ліквідує небажані функціональні залежності між атрибутами та забезпечує мінімальне дублювання даних за рахунок раціонального групування атрибутів.

В цьому підрозділі необхідно навести необхідні теоретичні відомості та проілюструвати процес нормалізації на конкретних

прикладах із БД, що проектується. Якщо в процесі розробки схеми БД були враховані всі вимоги нормалізації, то слід довести, що схема відповідає вимогам 4НФ.

Крім того, необхідно навести екранну копію схеми бази даних, як показано на рисунку 3.1.

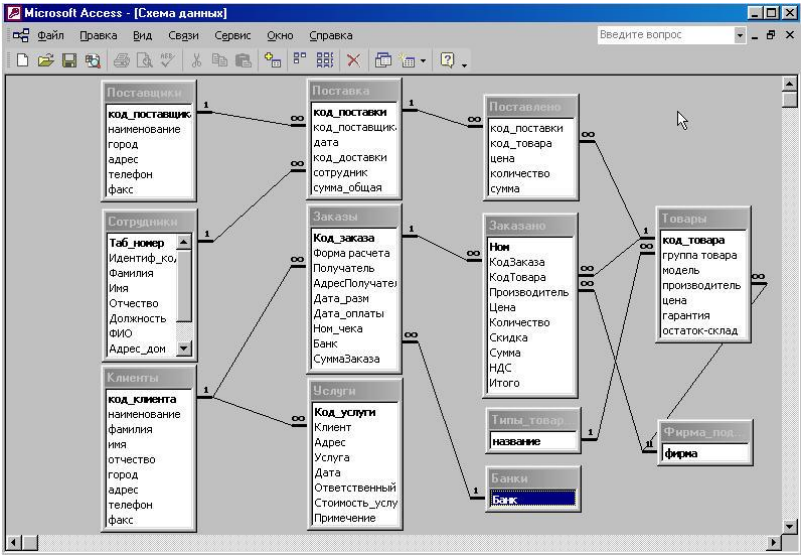

Рисунок 3.1 – Приклад схеми БД

3.3.3 Проектування додатків

Після розробки схеми бази даних приступають до проектування додатків, до яких, перш за все, відносяться запити. Крім того, це можуть бути макроси та (або) програми, що написані, наприклад, на мові Visual Basic.

Запити зазвичай проектуються за допомогою мови SQL, вбудованої майже в усі сучасні СКБД.

На якість спроектованих запитів багато в чому впливає досвід розробника, але можна сформулювати деякі корисні правила, за допомогою яких можуть бути оптимізовані запити. Як основні стратегії оптимізації можна запропонувати такі:

✓ Виконувати операцію селекції як можна раніше. Це пере-

творення запитів більше, чим щось інше, впливає на економію часу виконання, оскільки воно зменшує кількість проміжних обчислень.

✓ Доцільно спочатку обробити файли перед виконанням з`єднання (або, що еквівалентно, декартовому добутку з наступною селекцією). За процедуру попередньої обробки доцільно використовувати сортування файлів та їх індексацію.

✓ Шукати загальні підвирази у виразах. Якщо результат загального підвиразу являє собою невелике відношення, яке можна прочитати із зовнішньої пам`яті за значно менший час, ніж потрібно для його обчислення, доцільно спочатку обчислити цей загальний підвираз.

✓ Збирати каскади селекції та проекції. Послідовність таких операцій, кожна з яких вимагає тільки один операнд, може виконуватися одночасно, за один перегляд файлу операнду.

✓ Комбінувати проекції попередніми або наступними двомісними операціями. Немає необхідності робити перегляд файлу тільки для вилучення певних полів, або при першому використанні.

✓ Комбінувати деякі селекції з попереднім декартовим добутком та виконувати замість них з`єднання. Поєднання може бути виконано ефективніше декартова добутку тих же самих відношень.

При оформленні цього розділу необхідно всі спроектовані запити класифікувати за функціональним призначенням:

– вибірка;
– вилучання;
– додавання;
– оновлення.

Крім того, для кожного запиту слід навести:

– призначення;
– атрибути яких таблиць використовуються;
– умови виконання;
– текст запиту на мові SQL.

Якщо в системі крім запитів використовуються інші додатки, то для кожного з них треба зробити детальний опис, як і для запитів. У випадку, коли текст програми займає місця більше сторінки, то його слід наводити в додатку до ПЗ.

3.4 Проектування інтерфейсу користувача

База даних є невід'ємною складовою частиною будь-якої автоматизованої інформаційної системи. Користувачами таких систем зазвичай є люди, далекі від розуміння тонкощів обчислювальної техніки. Тому виникає задача розробки зручного інтерфейсу користувача для роботи з системою.

Під інтерфейсом користувача розуміється комплекс засобів, які надаються програмою для керування нею та перегляду результатів її роботи.

Для баз даних інтерфейс складається з форм вводу й редагування даних, а також звітів, які після перегляду на екрані можуть бути надруковані.

Перш ніж приступати до реалізації форми (або звіту), важливо ретельно спроектувати її (або його) макет. Можна дати деякі корисні рекомендації зі створення макетів будь-яких форм і звітів. Зокрема, подібний макет повинний включати:

– змістовна назва;
– ясні та зрозумілі інструкції;
– логічно обґрунтовані угруповання та послідовності полів;
– візуально привабливий вигляд вікна форми або поля звіту;
– погоджену термінологію та скорочення;
– погоджене використання кольорів;
– візуальне виділення простору та границь полів уведення даних;
– зручні засоби переміщення курсору;
– засоби виправлення окремих помилкових символів і цілих полів;
– засоби виводу повідомлень про помилки при введенні неприпустимих значень;
– особливе виділення необов'язкових для введення полів;
– засоби виводу пояснювальних повідомлень із описом полів;
– засоби виводу повідомлення про закінчення заповнення форми.

Розглянемо детальніше ці рекомендації.

Змістовна назва

Інформація в назві повинна ясно та недвозначно ідентифікувати призначення звіту або форми.

Ясні та зрозумілі інструкції

В інструкціях повинна застосовуватися звична для користувачів термінологія. Інструкції повинні бути короткими, а на випадок необхідності надання додаткової інформації слід передбачити спеціальні довідкові екрани. Інструкції варто записувати в стандартному форматі, дотримуючись єдиного граматичного стилю.

Логічно обґрунтовані угруповання та послідовності полів

Логічно зв'язані поля в звіті або формі варто розташовувати разом, причому їхня послідовність повинна бути логічно обґрунтованою та погодженою.

Погоджена термінологія та скорочення

Скрізь повинні використовуватися тільки знайомі та зрозумілі терміни чи скорочення, обрані зі заздалегідь погодженого списку.

Візуально привабливий вигляд вікна форми або полів звіту

Форма чи звіт повинні мати привабливий зовнішній вигляд і являти собою гармонічне сполучення полів або груп полів, рівномірно розподілених на поверхні форми/звіту.

При цьому у формі/звіті не повинно бути ділянок з дуже малою або занадто великою концентрацією полів. Крім того, поля потрібно розміщувати через регулярні інтервали та вирівнювати їх по вертикалі та горизонталі.

Якщо екранна форма має якесь еквівалентне представлення на папері, то їхній зовнішній вигляд повинний бути узгоджений.

Погоджене використання кольорів

Для поліпшення зовнішнього вигляду форми чи звіту можна використовувати кольорове оформлення. Крім того, виділення кольором може застосовуватися для найважливіших полів або повідомлень.

Для досягнення оптимального результату колір варто використовувати узгоджено та продумано. Наприклад, у формах поля із білим тлом можуть позначати поля введення, а поля із синім тлом - поля з даними, призначеними тільки для відображення на екрані.

Візуальне виділення простору та границь полів уведення даних

Користувач повинний бути візуально проінформований про загальний простір, доступний для введення даних у кожному з полів. Це дозволить йому ще до введення даних вибрати для них найбільш принадну форму представлення.

Зручні засоби переміщення курсору

Користувач повинний легко визначати, які операції доступні йому для переміщення курсору у формі чи звіті. Зазвичай для подібних цілей використовуються клавіші табуляції, клавіші зі стрілками або покажчик миші.

Засоби виправлення окремих помилкових символів і цілих полів

Користувач повинний легко визначати, які саме операції доступні йому для виправлення помилки, допущеної при введенні даних. Для цієї мети зазвичай використовуються найпростіші механізми, подібні до натискання клавіші <Backspace> або повторному введенню поверх помилкових символів.

Засоби виводу повідомлень про помилки при введенні неприпустимих значень

При введенні в поле неправильних даних програма повинна виводити повідомлення про помилку. Це повідомлення повинне інформувати користувача про допущену помилку та вказати діапазон припустимих значень.

Особливе виділення необов'язкових для введення полів

Необов'язкові для введення поля повинні бути явно відзначені за допомогою відповідного напису або виділення особливим кольором. Подібні поля варто розташовувати після обов'язкових для введення полів.

Засоби виводу пояснювальних повідомлень з описом полів

Коли користувач розміщує курсор миші в чергове поле, то в деякому стандартному місці (наприклад, у рядку стану даного вікна)

варто вивести інформацію про це поле.

Засоби виводу повідомлень про закінчення заповнення форми

Користувач повинний ясно уявляти собі, коли процес заповнення форми буде закінчений. Однак завершення цього процесу не повинне бути автоматичним - доцільно виводити попереджуюче повідомлення, щоб при необхідності користувач зміг ще раз переглянути введені ним дані.

У цьому розділі ПЗ слід навести зразки всіх екранних форм, що використовуються в системі, наповнені реальними даними. Послідовність слідування форм повинна відповідати їх рівню вкладеності:

- Заставка системи;
- Головна форма;
- форми першого рівня, наприклад, Довідники, Звіти, Замовлення, Постачання;
- форми наступного рівня, наприклад, Довідник одиниць виміру, Довідник груп товарів, тощо.

На Заставці системи можна розмістити у стислій формі відомості про організацію, де буде експлуатуватися система, призначення системи, логотип організації, відомості про розробника системи.

У якості головної форми можна використовувати кнопкову форму або сукупність укладок. Приклади форм наведені у Додатку Д.

Для пов’язаних таблиць треба обов’язково використовувати підлеглі форми. Це сприяє кращому розумінню користувачем інформації, а також підвищує ступінь достовірності даних.

3.5 Робота користувача з системою

Цей розділ ПЗ слід почати з меню системи, приклад якого показаний на рисунку 3.2.

Далі слід навести детальні інструкції користувачеві відносно роботи з системою з посиланнями на відповідні форми. Крім того, якщо є якісь обмеження на введення даних, їх необхідно також надати

користувачу.

Також слід навести вимоги з боку системи до складу її програмного та технічного забезпечень.

Якщо є якісь особливості, що стосуються інсталяції системи або її експлуатації, їх теж треба вказати.

В цьому ж розділі необхідно описати методи та засоби захисту інформації, що підтримуються в системі.

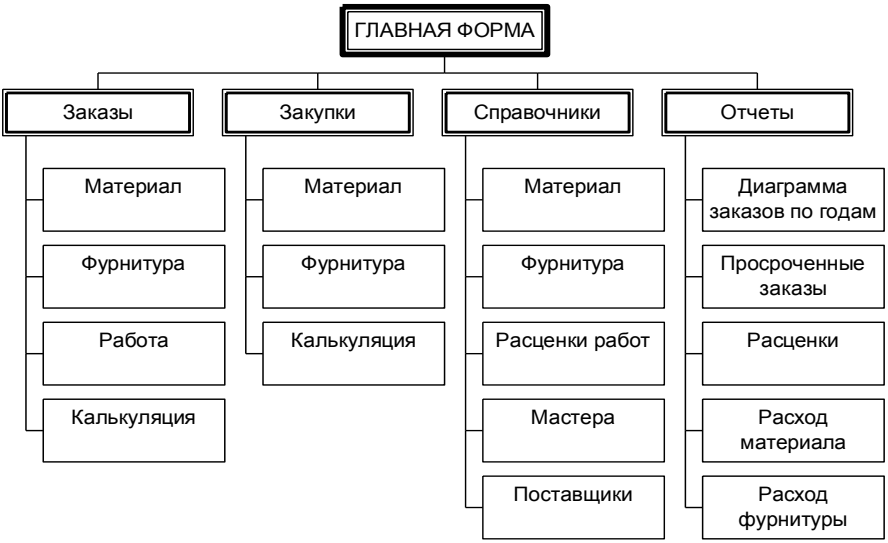

Рисунок 3.2 – Зразок меню системи

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Берлинер Э.М., Глазырин Б.Э., Глазырина И.Б. Офис от Microsoft. Москва: ABF, 1997. - 751 с.
2. Боуман Д, Эмерсон С., Дарновски М. Практическое руководство по SQL. - Киев: Диалектика, 1997. – 320 с.
3. Брункшир Дж. Введение в компьютерные науки. Общий обзор: Пер. с англ. – М.: Вильямс, 2001. –688 с.
4. Гарнаев А. Самоучитель VBA. – СПб.: БХВ – Санкт-Петербург, 1999. –512 с.
5. Гетц К., Джилберт М. Программирование в Microsoft Office .- К.: БХВ, 2000. –384 с.
6. Гетц К., Литвин П., Гилберт М. Access 2000. Руководство разработчика. Том 1. Настольные приложения. К.: BHV, 2000. – 1264 с.
7. Горев А., Аханян Р., Макашарипов С. Эффективная работа с СУБД – СПб: Питер, 1997. – 704 с.
8. Грабер М. Справочное руководство по SQL. - М.: Лори, 1997. – 291 с.
9. Дейт К. Введение в системы баз даннях: Пер. с англ. – К.:Диалектика,1998.-781 с.
10. Джексон Г. Проектирование реляционных баз данных для использования с ЭВМ: Пер. с англ. - М.: Мир, 1991. - 252 с.
11. Дженнингс Р. Руководство разработчика баз дананных на Visual Basic 6. – К.; М.;СПб.: Издательский дом "Вильямс", 1999. – 976 с.
12. Карпов Б. Microsoft Access 2000: Справочник. – СПб.: Питер, 2000. – 416 с.
13. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2001. – 304 с.
14. Конноли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика: Пер. с англ. – М.: Издательский дом "Вильямс", 2000. – 1120 с.
15. Кузьменко В.Г. Visual Basic 6. – М.: "Бином-Пресс",2003. –432 с.
16. Методичні вказівки до лабораторних робіт "Вивчення пакетів MS Excel і MS Access" (частина 1) з дисципліни "Організація баз даних і знань" для студентів спеціальності 8.080403 “Програмне забезпечення автоматизованих систем" (всіх форм навчання) /Укл.: С.К.Корнієнко. – Запоріжжя: ЗНТУ,2003.– 70 с.

17. Методичні вказівки до лабораторних робіт "Вивчення пакетів MS Excel і MS Access" (частина 2) з дисципліни "Організація баз даних і знань" для студентів спеціальності 8.080403 “Програмне забезпечення автоматизованих систем" (всіх форм навчання) /Укл.: С.К.Корнієнко. – Запоріжжя: ЗНТУ,2003.– 62 с.
18. Назаров С.В., Мельников П.П. Программирование на Visual Basic 6. – М.: Финансы и статистика,2002. – 320 с.
19. Нортон П., Андерсон В. Разработка приложений в Access 97 в подлиннике. – СПБ.:БХВ – Санкт-Петербург, 1999. –656 с.
20. Праг К.Н., Ирвин М.Р. Библия пользователя Access 2000. – М.: Вильямс, 2001. – 1040 с.
21. Сайлер Б., Споттс Д. Использование Visual Basic 6. Специальное издание. - К.; М.;СПб.: Издательский дом "Вильямс", 2003. – 832 с.
22. Visual Basic 6. - СПб.: БХВ – Санкт-Петербург, 2003. –992 с.

Додаток А

Приклад складення реферату

РЕФЕРАТ

ПЗ: 60с., 20 рис., 12 табл., 2 додатки, 5 джерел.

Об’єкт проектування – автоматизована інформаційно-пошукова система "Ательє", що працює під управлінням СКБД MICROSOFT ACCESS 2000.

Мета курсового проекту – розробка системи, що забезпечує автоматизацію обліку комерційної діяльності малого підприємства з пошиву верхнього одягу.

Розроблена система дозволяє вирішувати такі задачі:

- облік клієнтів;
- облік постачальників;
- облік замовлень;
- облік закупівель;
- автоматична калькуляція замовлень;
- автоматична калькуляція закупівель;
- генерація звітової документації;
- підведення підсумків діяльності за вказаний період;
- ведення довідкової системи.

Автоматизована система була розроблена з використанням операційної системи WINDOWS XP і СКБД ACCESS 2000.

Створений програмний продукт відповідає всім вимогам замовника та є доцільним для впровадження.

БАЗА ДАНИХ, СКБД, ІНТЕРФЕЙС, ЗАМОВЛЕННЯ, ЗАКУПІВЛЯ, ФОРМА, ЗАПИТ, ЗВІТ, МАКРОС

Додаток Б

Приклад змісту пояснювальної записки

ЗМІСТ

Завдання на проект..... 2
Реферат.....4
Перелік умовних позначень і скорочень.....6
Вступ.....7
1 Вибір і обґрунтування проектних рішень.....8
1.1 Технічне завдання.....8
1.1.1 Підстава для розробки.....8
1.1.2 Призначення розробки 8
1.1.3 Основні вимоги до системи, що розробляється.....8
1.2 Характеристика об’єкта проектування.....10
1.3 Обґрунтування структури системи, що проектується.....11
2 Концептуальне проектування предметної області.....12
3 Логічне проектування.....17
3.1 Вибір і обґрунтування інструментарію проектування.....17
3.2 Розробка схеми бази даних.....21
3.3 Нормалізація схеми бази даних.....25
3.4 Проектування запитів.....28
3.5 Проектування додатків.....34
3.6 Проектування звітів.....37
4 Інтерфейс користувача системи.....41
5 Методика роботи користувача з системою.....47
Висновки.....49
Перелік посилань.....50
Додаток А Зразки звітів.....51
Додаток Б Тексти програм.....59

Додаток В
Приклад оформлення технічного завдання

ЗАТВЕРДЖУЮ
зав. кафедрою ПЗ, професор

_____ В.І.Дубровін

“____” _____ 2014

АВТОМАТИЗОВАНА СИСТЕМА
“ОБЛІК”
ЛИСТ ЗАТВЕРДЖЕННЯ

Технічне завдання
13.02070849.00009-01-12-ЛУ
(Гнучкий магнітний диск)
4 листи

Керівник
_____ С.К.Корнієнко
“____” _____ 2014

Розробила
ст. гр. КНТ-413
_____ О.В. Сабліна
“____” _____ 2014

Нормоконтролер
_____ М.В. Калініна
“____” _____ 2014

2014 Літера

ЗАТВЕРДЖЕНИЙ
13.02070849.00009-01-12-ЛУ

АВТОМАТИЗОВАНА СИСТЕМА
“ОБЛІК”
ТИТУЛЬНИЙ ЛИСТ

Технічне завдання
13.02070849.00009-01-12
(гнучкий магнітний диск)
4 листи

2014 Літера

Вступ

Автоматизована інформаційно-пошукова система (АІПС) “ОБЛІК” призначена для автоматизації обліку ведення комерційних операцій торгово-виробничої фірми, що займається продажем обчислювальної техніки та наданням комп'ютерних послуг.

1 Підстава для розробки

Підставою для розробки є завдання на курсовий проект на тему “Автоматизація обліку комерційної діяльності торгово-виробничої фірми”, затверджене на засіданні кафедри ПЗ, протокол № від 200_ р.

2 Призначення розробки

АІПС призначена для обліку руху товарів на складі, одержання інформації про постачальників, клієнтів, постачаннях і замовленнях, оформлення відповідної звітної документації.

3 Основні вимоги до системи, що розробляється

3.1 Вимоги до функціональних характеристик

Початковими даними є таблиці “Товари”, “Постачання”, “Поставлено”, “Клієнти”, “Постачальники”, "Замовлення".

Вихідними даними програми є результати виконання запитів і звіти.

При введенні початкових даних програма повинна попередньо перевіряти правильність їх введення, щоб уникнути некоректної інтерпретації даних.

Інтерфейс користувача повинен забезпечити просте й ефективне спілкування користувача з програмним продуктом.

3.2 Вимоги до надійності

Програма повинна розпізнавати ситуації нерозуміння програмою користувача (користувачем програми), а також обробку збійних ситуацій.

3.3 Умови експлуатації

Програма може бути записана як на гнучкому, так і на твердому магнітному диску.

Експлуатація програмного продукту здійснюється відповідно до експлуатаційної документації на розроблену програму, що відповідає стандартам і містить інформацію, необхідну для освоєння та експлуатації програмного продукту.

3.4 Вимоги до складу і параметрів технічних засобів

Як апаратно-технічні засоби для експлуатації системи повинен використовуватися IBM-сумісний комп'ютер з характеристиками не нижче: процесор Duron 800 MHz, RAM 128 Мб, Video 32 Мб, HDD 20 Гб.

3.5 Вимоги до маркірування й упакування

Програма повинна поставлятися на гнучких дисках 3’5. На упакуванні повинна бути назва програми – “ОБЛІК”.

3.6 Вимоги до транспортування та збереження

Вимоги до транспортування та збереження аналогічні вимогам, пропонованим до гнучких дисків.

3.7 Вимоги до програмної документації

Система повинна поставлятися з “Інструкцією користувача”.

Додаток Д

Приклади форм

Рисунок Д.1 – Приклад Заставки до системи

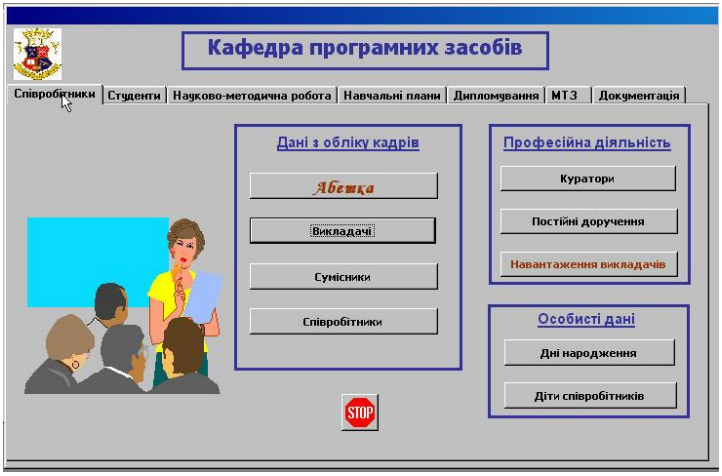

Рисунок Д.2 – Приклад використання сукупності вкладок

Рисунок Д.3 – Приклад використання сукупності вкладок

Рисунок Д.4 – Приклад використання кнопкової форми

Рисунок Д.5 – Приклад використання кнопкової форми

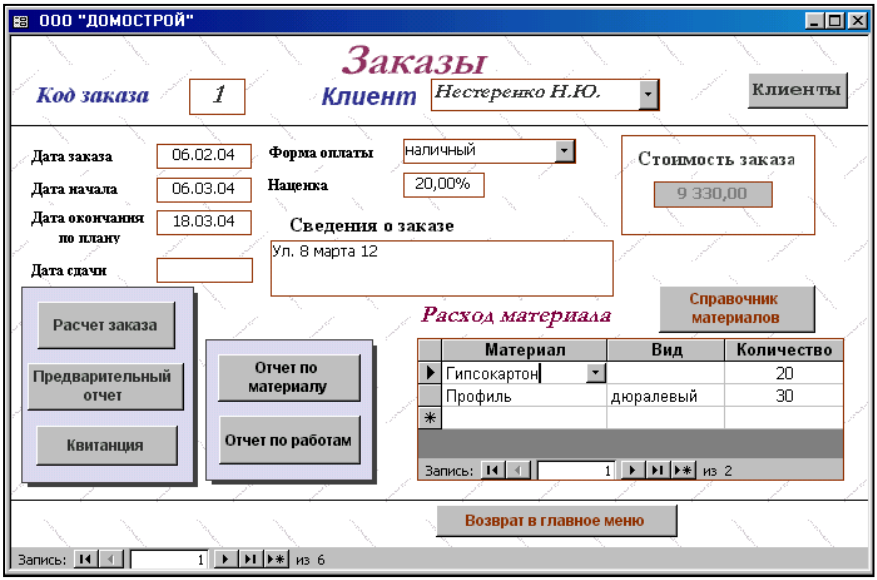

Рисунок Д.6 – Приклад використання зв’язаних форм

Додаток Ж

Словник основних термінів

Адміністратор бази даних – особа, яка відповідає за виконання функцій *адміністрування* бази даних, тобто координацію дій по збору відомостей, проектуванню та експлуатації бази даних, а також по забезпеченню захисту даних.

Альтернативний ключ – атрибут (або група атрибутів), які не співпадають із первинним ключем і унікально ідентифікують кожний рядок у таблиці.

Атрибут – інформаційне відображення властивостей об'єкту.

База даних – поіменована структурована сукупність взаємозв'язаних даних, які відносяться до конкретної предметної області та призначені для задоволення інформаційних потреб багатьох користувачів.

Безпека БД - захист від несанкціонованого доступу, зміни або руйнування даних.

Домен – набір значень елементів даних одного типу, який відповідає поставленим умовам.

Екземпляр сутності – опис конкретного об'єкту в наборі.

Зв'язок – функціональна залежність між сутностями.

Зв'язне відношення – відношення, яке зберігає ключі двох або більше об'єктних відношень, за якими встановлюється зв'язок між цими відношеннями.

Ключовий елемент даних – елемент, по якому можна визначити значення інших елементів даних.

Курсор (віртуальна таблиця) – об'єкт, який не містить власних даних. Це свого роду віртуальна таблиця, що не існує як незалежний об'єкт в базі даних і вміст якої береться з інших таблиць шляхом виконання запиту.

Нормалізація відношень – процес побудови оптимальної структури таблиць і зв'язків у реляційній базі даних.

Об'єкт – елемент інформаційної системи, інформація про який зберігається.

Об'єктне відношення – відношення, яке зберігає дані про об'єкти (екземпляри сутності) предметної області.

Первинний ключ – атрибут (або група атрибутів), які

однозначно ідентифікують кожний рядок у таблиці.

Посилальна цілісність – забезпечення співвідношення значень зовнішнього ключа екземпляра дочірній сутності значенням первинного ключа в батьківській сутності.

Предметна область – частина реального світу, яка представляє цікавість для даного дослідження.

Словник даних – централізоване сховище відомостей про об'єкти, елементи даних, які їх складають, взаємозв'язки між об'єктами, їх джерела, значення, використання та формати представлення.

Ступінь відношення – кількість атрибутів (стовпців) відношення.

Сутність – це збиральне поняття, деяка абстракція реально існуючого об'єкта навколишнього світу, процесу або явища.

Таблиця (відношення) – деяка регулярна структура, яка складається з кінцевого набору однотипних записів.

Тип сутності – набір однорідних об'єктів, які виступають як єдине ціле.

Транзакція – це логічна одиниця роботи, яка виконується без порушення логічної цілісності бази даних. Якщо в процесі виконання транзакції виникла помилка виконання, то система, яка підтримує процес транзакції, гарантує повернення до первинного стану.

Тригер – попередньо визначена дія або послідовність дій, які автоматично здійснюються під час виконання операцій оновлення, додавання або вилучання даних.

ODBC (Open Database Connectivity) – відкритий доступ до баз даних – загальне визначення мови та набір протоколів, які дозволяють клієнтському додатку працювати з командами та функціями, що підтримуються сервером.

OLE (Object Linking and Embedding) – зв'язування та впровадження об'єктів – технологія, яка дозволяє використовувати в додатку об'єкти, розроблені в іншому додатку. OLE-об'єктами можуть бути звуки, рисунки, діаграми, відеокліпи тощо.

OLE-сервер – програма, яка може надавати іншим програмам можливість використання своїх об'єктів.

SQL (Structured Query Language) – мова структурованих запитів, офіційний стандарт мови для роботи з реляційними базами даних.