

МІНІСТЕРСТВО ОСВІТИ УКРАЇНИ  
ВІННИЦЬКИЙ ДЕРЖАВНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

## НАВЧАЛЬНИЙ ПОСІБНИК "ІНТЕРФЕЙС КОРИСТУВАЧ-КОМП'ЮТЕР"

Для студентів бакалаврського напрямку 6.0804 -  
"Комп'ютерні науки", спеціальності 7.080403 - "Програмне  
забезпечення автоматизованих систем"

Усі цитати, цифровий,  
фактичний матеріал та  
бібліографічні відомості  
перевірені, написання  
одиниць відповідає  
стандартам.

Зауваження рецензентів  
враховані.

Автори: А.М. Петух  
В.П. Майданюк

Вимогам, які пред'являються до  
інструктивно - методичної  
літератури відповідає.

До друку і в світ дозволяю на  
підставі п.2.15 "Єдиних правил".

Проректор з навчальної роботи  
В.О. Леонтєв

Затверджено на засіданні  
кафедри прикладної математики і  
обчислювальних систем, протокол  
N \_\_\_\_\_ від "\_\_\_" \_\_\_\_\_

Зав. кафедри ПМОС

\_\_\_\_\_ А.М. Петух

**МІНІСТЕРСТВО ОСВІТИ УКРАЇНИ**  
**ВІННИЦЬКИЙ ДЕРЖАВНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

**В.П. Майданюк    А.М. Петух**

# **ІНТЕРФЕЙС "КОРИСТУВАЧ-КОМП'ЮТЕР"**

**Вінниця ВДТУ 1999**

**МІНІСТЕРСТВО ОСВІТИ УКРАЇНИ**  
**ВІННИЦЬКИЙ ДЕРЖАВНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

**В.П. Майданюк    А.М. Петух**

**ІНТЕРФЕЙС "КОРИСТУВАЧ-КОМП'ЮТЕР"**

Затверджено на засіданні Ученої ради Вінницького державного технічного університету як навчальний посібник для студентів спеціальності "Програмне забезпечення автоматизованих систем"

Протокол №        від

**Вінниця ВДТУ 1999**

УДК 681.3.06

Інтерфейс "Користувач-комп'ютер": Навчальний посібник/  
В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с. Укр.  
Мовою/

У навчальному посібнику розглянуті питання організації інтерфейсу "людина-комп'ютер" в діалогових системах. Приведені особливості різних типів структур діалогу, значно увага приділена багатовіконним WIMP-інтерфейсам, які є основою інтерфейсу "людина-комп'ютер" сучасних операційних систем персональних комп'ютерів. Розглянуто основні концепції стандарту CUA фірми IBM по проектуванню інтерфейсу "людина-комп'ютер". Навчальний посібник містить лабораторний практикум, який допоможе студентам краще засвоїти викладений матеріал. Відомості наведені в навчальному посібнику можуть використовуватись як при самостійній роботі студентів, так і при виконанні курсових і дипломних робіт.

Навчальний посібник призначено для студентів спеціальності "Програмне забезпечення автоматизованих систем".

Бібліогр. 10 назв, іл 15

Рецензенти: В.П. Кожем'яко, д.т.н., проф.  
Р.Н. Кветний, д.т.н., проф.  
А.Т. Теренчук, к.т.н., доц.

**МІНІСТЕРСТВО ОСВІТИ УКРАЇНИ**  
**ВІННИЦЬКИЙ ДЕРЖАВНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

Навчальне видання

Володимир Павлович Майданюк  
Анатолій Михайлович Петух

Інтерфейс "Користувач - комп'ютер"

Навчальний посібник

Вінниця ВДТУ 1999

Редактор В.О. Дружиніна  
Коректор З.В. Поліщук

## ЗМІСТ

ВСТУП.....	4
1. ЗНАЧЕННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧ-КОМП'ЮТЕР .....	5
1.1. Інтерфейс користувач-комп'ютер та його зручність .....	5
1.2. Компоненти інтерфейсу користувач-комп'ютер.....	7
1.3. Стратегії розробки інтерфейсу користувач-комп'ютер .....	10
1.4. Класифікація повідомлень в діалозі.....	11
1.5. Критерії оцінки якості діалогу.....	12
КОНТРОЛЬНІ ПИТАННЯ .....	14
2. СТРУКТУРИ ДІАЛОГУ .....	15
2.1. Структура діалогу типу запитання-відповідь.....	15
2.2. Структура діалогу типу "меню" .....	16
2.3. Структура діалогу на основі екранних форм .....	17
2.4. Структура командного типу діалогу.....	18
2.5. Інтелектуальні інтерфейси.....	19
КОНТРОЛЬНІ ПИТАННЯ .....	21
3. ОСНОВИ ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА ....	22
3.1. Підтримка користувача.....	22
3.2. Розміщення інформації на екрані .....	23
3.3. Час відповіді системи .....	24
3.4. Види адаптації в діалогових системах .....	25
3.5. Опис діалогу .....	27
КОНТРОЛЬНІ ПИТАННЯ .....	30
4. БАГАТОВІКОННІ WIMP-ІНТЕРФЕЙСИ.....	31
4.1. Основні аналоги .....	31
4.2. Вікна .....	33
4.3. Переваги і недоліки WIMP-інтерфейсів.....	35
4.4. Критерії проектування WIMP-інтерфейсів.....	37
КОНТРОЛЬНІ ПИТАННЯ .....	38
5. СТАНДАРТ ФІРМИ ІВМ ПО ПРОЕКТУВАННЮ .....	39
ІНТЕРФЕЙСІВ КОРИСТУВАЧА.....	39
5.1. Аналоги і принципи CUA .....	39
5.2. Панелі і вікна в CUA .....	40
5.3. Уніфіковані дії діалогу .....	43
5.4. Допоміжні функції .....	45
КОНТРОЛЬНІ ПИТАННЯ .....	49
6. ЛАБОРАТОРНИЙ ПРАКТИКУМ .....	50
6.1. Проектування інтерфейсу користувача у середовищі Borland Delphi.....	50
6.2. Методичні вказівки .....	57
6.3. Завдання.....	60
6.4. Етапи виконання роботи .....	61
ЛІТЕРАТУРА.....	64

## **ВСТУП**

У даному навчальному посібнику розглядаються способи організації інтерфейсу людина-комп'ютер. Питання, пов'язані з проектуванням зручних засобів спілкування людини з комп'ютером, мають велике значення в зв'язку з широким застосуванням персональних комп'ютерів і значним впливом інтерфейсу людина-комп'ютер на продуктивність праці користувачів. Особливо це відноситься до непостійних користувачів або не пов'язаних з обчислювальною технікою.

У посібнику докладно розглянуті основні типи організації діалогу між комп'ютером та користувачем, а саме такі як, "запитання - відповідь", діалог командного типу, "меню", екранні форми, багатовіконні WIMP-інтерфейси. Приведені основні положення стандарту фірми IBM (CUA), який визначає правила проектування текстових і графічних інтерфейсів користувача. Посібник містить також лабораторний практикум, який допоможе студентам краще засвоїти викладений матеріал.

На думку авторів наведені в навчальному посібнику відомості можуть бути використані студентами, а також магістрами та аспірантами в їх самостійній роботі.

Матеріал навчального посібника підготували А.М. Петух вступ, розділ 1, В.П. Майданюк - розділи 2, 3, 4, 5, 6.

Автори висловлюють щире подяку рецензентам, доброзичливе ставлення та поради яких сприяли покращенню змісту та структури навчального посібника.

# 1. ЗНАЧЕННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧ-КОМП'ЮТЕР

## 1.1. Інтерфейс користувач-комп'ютер та його зручність

Раніше при розробці комп'ютерних систем основна увага приділялась точності роботи цих систем. Точність означає, що при надходженні на вхід системи заданих значень на її виході отримують задані результати. Значно менше уваги зверталось на зручність роботи користувача з системою [1,2].

Користувач автоматизованої обчислювальної системи має право очікувати не тільки точних результатів обробки, але і зручності в використанні системи. Тіло людини (користувача) - це механізм, який працює в рамках визначених обмежень і допусків. Очам людини необхідно, щоб образи мали визначений розмір, рівень яскравості, контрастності і розміщувались на зручній відстані. Деякі кольори сприймаються краще інших. Необхідно відмітити і обмеження мозку людини. У людини велика довготривала пам'ять і дуже обмежена короткочасна пам'ять. Подібно буферам обчислювальної системи ця пам'ять може легко перевантажуватись. Люди можуть розширювати цю пам'ять за допомогою записів на листках паперу, застосовуючи нові моделі роботи або пристосовуючись до нового способу роботи. Однак така адаптація може приводити до стресів, а як результат до неприйняттого рівня помилок.

Критерій зручності став визначальним в останні роки в зв'язку з широким застосування персональних комп'ютерів з діалоговими режимами роботи в різних галузях народного господарства, побуті, що привело до зростання кількості користувачів неспеціалістів в області інформатики та обчислювальної техніки. Згідно з класифікацією фірми IBM [2] розрізняють такі типи користувачів:

- системний програміст;
- програміст;
- кінцевий користувач.

Кінцеві користувачі - це люди, які хочуть більш ефективно і економно вирішувати свої завдання, використовуючи комп'ютерні засоби. Частіше всього вони не є спеціалістами в області інформатики. Досвід показує, що ці користувачі можуть ефективно



вирішувати на машині свої завдання лише в умовах наявності активної допомоги зі сторони системи на всіх етапах вирішення завдання і спрощення методів взаємодії з обчислювальною системою [3].

Зрозуміло, що сучасні обчислювальні системи повинні орієнтуватись на таких користувачів в першу чергу, оскільки більшість користувачів персональних комп'ютерів є кінцевими користувачами. Використовуючи в подальшому термін користувач, ми маємо на увазі кінцевих користувачів.

Таким чином, з точки зору користувача діалогової системи - система є зручною для користувача, якщо інтелектуальні зусилля користувача, необхідні для розуміння дій системи і реакції на них мінімальні. Можливо сформулювати ряд вимог до системи з точки зору зручності:

1. Поведінка системи по відношенню до користувача повинна бути гнучкою, тобто користувач не повинен діяти строго визначеним способом.
2. Система повинна вміти розрізняти користувача і пристосовуватись до нього.
3. Поведінка системи повинна бути зрозумілою користувачеві.
4. Система завжди повинна бути готова допомогти користувачеві.
5. Для використання системи не потрібні спеціальні навички та додаткове навчання.
6. Не потрібно зловживати здатністю людини до навчання під час роботи з системою.
7. Система повинна реагувати на порушення взаємодії з користувачем, обумовлені властивостями людини і приймати запобіжні заходи проти цих порушень.

Інтерфейс користувач-комп'ютер включає всі ті аспекти обчислювальної системи, з якими безпосередньо взаємодіє користувач.

## 1.2. Компоненти інтерфейсу користувач-комп'ютер

Основою будь-якої діалогової системи є зручний інтерфейс "користувач-комп'ютер". Він забезпечує зв'язок між користувачем і процесом, який виконує деяке завдання. З точки зору програмного забезпечення в склад інтерфейсу входять дві компоненти (рис. 1.1):

- набір процесів введення-виведення;
- процес діалогу.

Процеси введення-виведення служать для того, щоб прийняти від користувача або передати йому дані через різні фізичні пристрої. З кожним пристроєм зв'язаний свій процес введення-виведення, задача якого прийняти дані від відповідного пристрою і перетворити їх в внутрішній формат з яким можуть працювати "процеси діалогу". Наприклад, прийняти від клавіатури скан-код клавіші "5" (00000110) і перетворити його в двійкове представлення символу в коді ASC11 (00110101). Якщо користувач вирішив скористатись мовним введенням, то слово "п'ять" також буде представлено аналогічним чином. Типовими пристроями введення-виведення сучасних комп'ютерів є такі:

### Пристрої виведення

- оперативна текстова і графічна інформація: монохромні і кольорові дисплеї на базі ЕПТ (електронно-променева трубка);
- тверда копія: алфавітно-цифрові друкувальні пристрої (виведення тексту); лазерні принтери (текстовий і графічний вивід); графопобудовувачі (графічний вивід);
- звукове виведення: синтезатори мови (мовне виведення); звукогенератори ("музикальне виведення").

### Пристрої введення

- введення довільних даних користувачами: клавіатура (текстове введення); планшети (графічне введення);
- автоматичне збирання інформації: пристрої зчитування документів, сканери;

- позиціонування і вибір: світлове перо, сенсорний екран, маніпулятори типу "миша", "джойстик", шар;
- пристрої мовного введення і машинного зору.

Для більшості застосувань розробка фізичного інтерфейсу включає в себе вибір необхідних пристроїв і створення на їх основі автоматизованих робочих місць. При виборі пристроїв враховуються такі фактори:

- зміст і формат даних, що обробляються;
- об'єми введення-виведення;
- обмеження, які вводяться користувачем і робочим середовищем;
- обмеження, пов'язані з іншими апаратними і програмними засобами, які використовуються в системі.

Процеси діалогу - це механізм обміну інформацією, який можна розглядати як оболонку, яка включає всі процеси, які входять в систему для виконання визначеного завдання.

Діалог між людиною і комп'ютером можна визначити як обмін інформацією між обчислювальною системою і користувачем, який виконується за допомогою інтерактивного терміналу за визначеними правилами.

Діалог користувач-ЕОМ (електронно-обчислювальна машина) можливо розглядати з двох точок зору. У відповідності з першою ("технічною") для забезпечення діалогу достатньо організувати лише швидкий обмін повідомленнями в режимі безпосередньої взаємодії з ЕОМ. При цьому час очікування суб'єктом чергової відповіді ЕОМ не повинен порушувати процес його мислення. Друга точка зору - під діалогом розуміють спілкування між користувачем і ЕОМ на природній мові. Ця точка зору характерна для розробників систем штучного інтелекту [1].

Метою даної роботи є вивчення особливостей розробки структур діалогу, які базуються на першому підході, оскільки поки що обчислювальна потужність ЕОМ недостатня для реалізації і широкого застосування діалогу на природній мові.

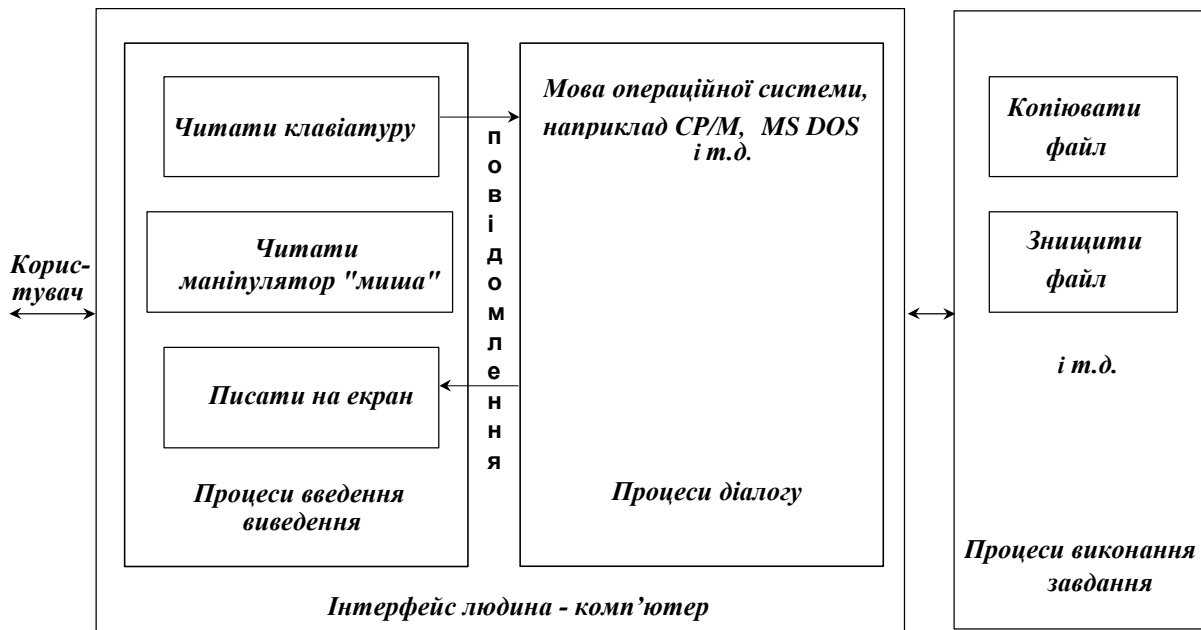


Рис. 1.1. Складові частини інтерфейсу людина-комп'ютер на прикладі операційної системи

Задачі процесів діалогу такі:

- визначити завдання, яке користувач покладає на систему;
- прийняти логічно-зв'язані вхідні дані від користувача і розмістити їх в змінних відповідного процесу в необхідному форматі;
- виклик процесу виконання необхідного завдання;
- виведення результатів обробки по закінченню процесу в придатному (зручному) для користувача форматі.

Незалежно від граматики і способу ведення діалогу в його основі лежить такий цикл:

- явний або неявний запит на введення даних;
- введення даних через процес введення;
- перевірка вхідних даних.

Всі початкові дані, незалежно від формату в якому вони вводяться представляють результат явного або неявного вибору з набору прийнятних відповідей. Якщо діапазон вхідних даних невеликий, то перевірка виконується шляхом порівняння введеного користувачем повідомлення з допустимими іменами завдань. Якщо діапазон вхідних значень великий, то перевірку можна виконати

шляхом порівняння форматів вхідного повідомлення з заданим. Наприклад, в запиті

**Введіть дату (дд-мм-гг) :**

виводиться підказка, яка подає формат введення дати. В такому випадку перевірка виконується шляхом порівняння вхідного повідомлення з заданим форматом.

З точки зору користувача інтерфейс людина-комп'ютер - це основний процес, а процеси, які виконують завдання є неосновними (фоновими).

### **1.3. Стратегії розробки інтерфейсу користувач-комп'ютер**

При розробці інтерфейсу користувач-комп'ютер рекомендується придержуватись наступних правил (стратегій) [1,2]:

- розглядати інтерфейс людина-комп'ютер як окремий компонент системи;
- враховувати можливості апаратних і програмних засобів;
- послідовності, тобто бажано щоб в рамках хоча б однієї операційної системи всі програми мали б однаковий інтерфейс користувача;
- використовувати загальноприйняті в ергономіці рекомендації по взаємодії з машиною взагалі, хоча необхідно враховувати особливості представлення інформації на екрані в різних обчислювальних системах;
- розробник інтерфейсу повинен досконало знати задачу користувача. Він повинен розуміти не тільки обчислювальний процес, необхідний для вирішення задачі, але і оцінити дії користувача, направлені на досягнення мети задачі;
- необхідно залучати користувачів для практичної перевірки розроблених зразків інтерфейсу людина-комп'ютер з наступною доробкою до тих пір, поки не буде створено прийнятний продукт;
- передбачати засоби адаптації в рамках інтерфейсу, тобто він повинен налагоджуватись на потреби різних користувачів або на потреби одного користувача в різні періоди його роботи.

## 1.4. Класифікація повідомлень в діалозі

Повідомлення, якими обмінюються учасники під час діалогу виконують декілька функцій, необхідних для підтримки діалогу. Класифікація повідомлень, якими обмінюються людина і комп'ютер під час діалогу приведена на рис. 1.2. [1,4]



Рис. 1.2. Класифікація повідомлень у відповідності з функціями, які вони виконують

Підказка це вихідне повідомлення системи, яке спонукає користувача вводити дані. Наприклад, підказка "ім'я пристрою", зрозуміла для більшості мікрокомп'ютерних систем:

C:\ \_ ,

де С - ім'я дискового пристрою.

Повідомлення про помилку - це сигнал діалогового процесу про те, що подальше виконання роботи неможливе, оскільки він або викликаний процес виконання завдання не може опрацювати повідомлення, введене користувачем.

Вихідні дані - це дані, які повертає процес по закінченню обробки.

Повідомлення про стан системи - це інформація для користувача про те, що відбулося або що відбувається в системі.

Довідкова інформація виводиться в тих випадках, коли користувач не може відповісти на запит системи, тому що не розуміє сам запит або забув що необхідно вводити.

Вхідне керуюче повідомлення або команда вибирає процес для виконання певного завдання, а вхідні дані дають інформацію цьому процесу з якими структурами даних виконувати це завдання.

Діалог можна класифікувати з урахуванням формату вхідного повідомлення і гнучкості, яка дозволяє користувачеві вводити вхідні повідомлення, коли йому зручно. Розрізняють:

- діалог, який керується системою;
- діалог який керується користувачем.

Перший тип діалогу характеризується тим, що система жорстко задає, яке завдання можливо вибрати і які дані вводити. Це виконується за допомогою системи підказок користувачеві. Діалог, який керується користувачем - це діалог в якому ініціатива повністю належить користувачеві, тобто він може подати будь-яку команду на виконання необхідного завдання. Відповідальність за вірність введеної комбінації повністю лягає на користувача.

Формат, в відповідності з яким користувач вводить свої повідомлення називається граматиною діалогу. Існує декілька варіантів граматики діалогу:

- коди;
- ланцюжки ключових слів, подібних до програми;
- обмежена природна мова;
- природна мова.

Реалізація діалогу на природній мові пов'язана з найбільшими труднощами. Хоча існує декілька систем з подібною граматиною, практична цінність такого підходу викликає сумнів.

## **1.5. Критерії оцінки якості діалогу**

Основні критерії оцінки якості діалогу такі:

- природність;
- послідовність;
- ненадмірність;
- підтримка користувача;
- гнучкість.

Природний діалог - це такий діалог, який не спонукає користувача суттєво змінювати свої традиційні способи вирішення задач. Як мінімум це означає, що діалог повинен вестись на рідній мові користувача. Розробник повинен не допускати безглузлого порядку постановки питань. Фрази, по можливості, не повинні вимагати додаткових пояснень. Неприродний діалог часто є наслідком того, що розробник системи не знайомий з тими способами, якими вирішує задачу користувач.

Діалог, який відрізняється логічною послідовністю, гарантує, що користувач, який добре володіє роботою однієї частини системи, легко зможе оволодіти роботою інших частин даної системи. Стандартні відповіді повинні бути дійсно "стандартними". Наприклад, довідка отримується по натисканню однієї клавіші F1 в будь-якій точці діалогу, ALT X - завершення роботи будь-якої програми в даному операційному середовищі. Послідовність в використанні форматів даних означає, що аналогічні поля завжди будуть представлятися системою в одному і тому ж форматі. Наприклад, якщо поле для роботи з датою має формат дд/мм/рр, то всі аналогічні поля повинні мати такий же формат. Послідовність у розміщенні даних на екрані в різних ситуаціях гарантує, що користувачеві відомо, де на екрані шукати інструкції, повідомлення про помилки і т.д.

Ненадмірний (короткий) діалог вимагає від користувача введення тільки мінімуму інформації, необхідної для роботи системи. Немає необхідності в діалозі запитувати інформацію, яку можливо сформувати автоматично або яка була вже введена раніше, наприклад поточна дата. Вихідні повідомлення повинні містити тільки ту інформацію, яку запитує користувач.

Підтримка користувача в процесі діалогу - це міра допомоги, яку діалог надає користувачеві під час його роботи з системою.

Основні складові цієї підтримки такі:

- кількість і якість інструкцій, які є в системі;
- характер повідомлень про помилки;
- підтвердження будь-яких дій системи.

Гнучкість діалогу - це міра того, наскільки він відповідає різним рівням підготовки і продуктивності праці користувача. Це передбачає, що діалог повинен вміти адаптувати свою структуру і вхідні дані у відповідності з рівнем підготовки користувача.



## КОНТРОЛЬНІ ПИТАННЯ

1. Що таке точність обчислювальної системи?
2. Як можливо визначити "зручність"?
3. Що таке інтерфейс користувач-комп'ютер?
4. Компоненти інтерфейсу користувач-комп'ютер?
5. Поясніть призначення процесів введення-виведення.
6. Назвіть типові пристрої введення-виведення.
7. Визначити задачі процесів діалогу.
8. Назвіть основні стратегії розробки інтерфейсу користувач-комп'ютер.
9. Дайте класифікацію повідомлень в діалозі.
10. В чому різниця між підказкою і довідкою?
11. Поясніть призначення повідомлення про стан системи.
12. Дайте класифікацію діалогу з урахуванням вхідного повідомлення.
13. Дайте характеристику основних граматик діалогу?
14. Приведіть основні критерії оцінки якості діалогу.
15. Як ви розумієте "послідовність діалогу"?
16. Наведіть ознаки, які характеризують "природний" діалог.

## 2. СТРУКТУРИ ДІАЛОГУ

Традиційно виділяється чотири основних структури [1,2]:

- запитання -відповідь;
- меню;
- екранні форми;
- діалог на основі команд.

Кожна з цих структур ґрунтується на аналогії взаємодії між людьми. Ці чотири основні структури діалогу можна розглядати як різновидності структури запитання - відповідь. Їх можливо представити таким чином:

- меню : опереджувальна довідка;
- форма : опереджувальне запитання;
- команда : опереджувальне введення.

Не дивлячись на те, що більшість систем мають в своїй основі структури діалогу типу запитання-відповідь, меню або структуру на основі команд, рідко вдається побудувати діалог для всієї системи, використовуючи тільки одну структуру. Для різних частин діалогу необхідні різні структури в залежності від їх конкретних характеристик, тобто більшість діалогів ґрунтується на змішаних структурах.

### 2.1. Структура діалогу типу запитання-відповідь

Структура діалогу типу запитання-відповідь(Q&A) основана на аналогії з звичайним інтерв'ю. Цей тип діалогу відноситься до діалогу, який керується комп'ютером. В кожній точці діалогу система виводить в якості підказки одне запитання, на яке користувач дає одну відповідь. В залежності від отриманої відповіді система може вирішити, яке наступне запитання задавати. Якщо відповідь невірна система виводить повідомлення про помилку і знову виводить підказку. Цей процес повторюється до тих пір, поки не буде отримано прийнятну відповідь. Структура Q&A не гарантує мінімального об'єму введення, який оцінюється в кількості натискань клавіш. Однак, при відповідному підборі скорочень, надлишковість

може бути зменшена. Для прискорення введення системи з структурою Q&A повинні мати механізм випереджального введення відповідей. Хоча структура Q&A з появою потужних засобів відображення інформації дещо застаріла, вона має ряд переваг. Це такі:

- структура Q&A є проміжною структурою, яка може задовольнити вимоги різних користувачів і типів даних;
- вона достатньо гнучка і в достатній мірі забезпечує підтримку користувача;

Ця структура застосовується найбільш часто, коли на кожне запитання передбачена велика кількість відповідей, кожна з яких впливає на зміст наступного запитання. Тому структуру Q&A знайшла широке використання в експертних системах.

## 2.2. Структура діалогу типу "меню"

Структура діалогу типу "меню" характеризується тим, що у користувача є список можливих варіантів для введення, із якого

<i>Персонал</i>	<i>Клієнти</i>	<i>Меблі</i>	<i>Пристрої</i>	<i>Різне</i>	<i>Кінець</i>
-----------------	----------------	--------------	-----------------	--------------	---------------

а)

<i>Персонал</i>
<i>Клієнти</i>
<i>Меблі</i>
<i>Пристрої</i>
<i>Різне</i>
<i>Кінець</i>

1. *Персонал*
2. *Клієнти*
3. *Меблі*
4. *Пристрої*
5. *Різне*
6. *Кінець*

в)

б)

*Рис. 2.1. Приклади різних форматів "меню". а - меню в вигляді рядка даних; б - меню в вигляді блоку даних; в - меню в вигляді списку об'єктів, які вибираються указанням цифрових кодів*

необхідно вибрати ті, що потрібно. Популярність меню особливо виросла в зв'язку з появою інтерфейсу, що керується маніпулятором

типу "миша" і появою "віконної" техніки [5,6] виведення даних. Існують різні формати представлення меню на екрані. Деякі з них приведені на рис. 2.1.

Меню в вигляді блоку даних на екрані традиційний формат. Меню в вигляді рядка даних може появлятися вверху або внизу екрана і часто залишається в цій позиції протягом діалогу. У випадку, коли на екрані недостатньо місця, додаткові меню "випадають" або "вискакують" (POP-UP) на екран в поточній позиції екрана або витягуються (PULL-DOWN) з рядка меню. Ці меню зникають після вибору варіанта. Меню у вигляді піктограм - це множина блоків об'єктів вибору, розкиданих по екрану, які часто містять графічне представлення об'єктів роботи. Користувач діалогового меню може вибрати необхідний пункт вводячи текстовий рядок, який ідентифікує цей пункт; указуючи на нього безпосередньо, наприклад за допомогою маніпулятора типу "миша", або продиляючи список в режимі ролика і вибираючи з нього. Основні переваги і недоліки меню такі:

1. Меню використовується для введення значень даних, діапазон яких невеликий і їх можна згрупувати в ієрархію.
2. Меню природна структура при роботі з пристроями указання і вибору (наприклад маніпулятор типу "миша").
3. Функціональні клавіші можна налаштувати так, щоб кожна з них відповідала пункту меню.
4. Просто сформулювати повідомлення про помилки, оскільки прийнятні варіанти відповідей відображені на екрані.
5. Меню можуть використовувати користувачі будь-якої кваліфікації.

### **2.3. Структура діалогу на основі екранних форм**

Структура діалогу типу "екранна форма" характеризується тим, що система отримує від користувача всю інформацію зразу способом заповнення форми. Аналогом екранної форми є звичайна анкета. На практиці форми використовуються там, де облік якої-небудь діяльності вимагає введення достатньо стандартного набору даних. Користувач може редагувати відповіді, пропускати, повертатись назад. Комп'ютерна система може перевіряти кожне введене повідомлення зразу після введення або після заповнення всієї форми. Деякі характеристики цієї структури:

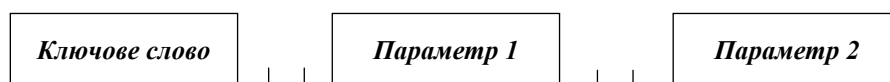
1. Працює швидше, ніж Q&A.
2. В порівнянні з "меню" може маніпулювати більш широким набором вхідних даних.
3. В меншій мірі підходить для роботи в режимі вибору варіантів, оскільки має послідовну, а не деревоподібну організацію.
4. Знаходить застосування в базах даних.
5. Може слугувати користувачам будь-якої кваліфікації.

## 2.4. Структура командного типу діалогу

Структура діалогу на основі командної мови відноситься до діалогу, що керується користувачем. Команда включає в себе постановку задачі, можливі пояснення і дані, необхідні для її вирішення. Діалог на базі команд працює в телетайпному режимі. Система нічого не виводить крім постійної підказки, Наприклад, підказка "ім'я пристрою", прийнята для більшості мікрокомп'ютерних ОС. Вона означає готовність системи до роботи. Кожну команду вводять з нового рядка і закінчують натисканням клавіші "ENTER". Наприклад,

**C:\ DIR <ENTER>**

Відповідальність за правильність команд лягає на користувача. Формат команди в загальному випадку приведений на рис. 2.2.



*Рис. 2.2. Формат команди*

Параметри в списку можна задавати в одному з двох форматів [1]:

- в вигляді позиційних параметрів;
- в вигляді ключових параметрів.

Значення параметра, який задається в позиційній формі, визначається за його місцем в командному рядку. Наприклад, в команді

## **COPY thisfile newfile**

перший параметр задає файл, який повинен копіюватись, а другий параметр, файл куди необхідно копіювати вміст першого. У випадку ключових параметрів кожне значення параметра позначається ідентифікатором, який задає його призначення. Наприклад, в команді

### **COPY SOURCE=thisfile DESTINATION=newfile**

ключові слова source (джерело) і destination (одержувач) визначають призначення кожного параметра. Недоліком позиційних параметрів є те, що помінявши місцями параметри можна отримати протилежний результат, а недоліком ключових параметрів є необхідність запам'ятовування великої кількості ключових слів.

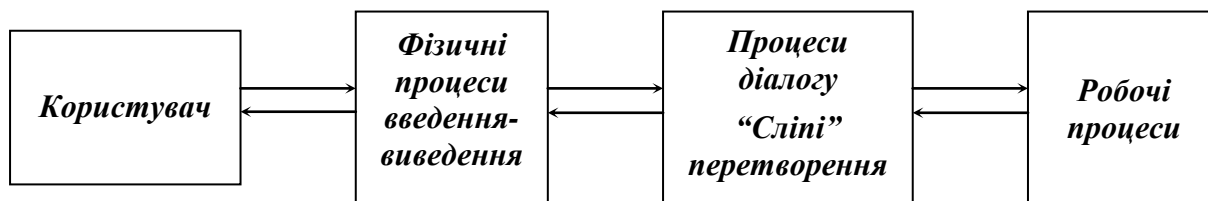
Список параметрів може містити ключі, які уточнюють спосіб інтерпретації команди. Ключі помічаються спеціальним ідентифікатором (наприклад "-", "/").

Структура діалогу на основі мови команд за своїми можливостями найшвидша і найгнучкіша із всіх структур діалогу, але для роботи в цьому режимі необхідна хороша підготовка користувача.

## **2.5. Інтелектуальні інтерфейси**

Всі інтерфейси, розглянуті вище були "сліпі". Пересилка інформації між фізичними процесами введення-виведення і користувачем включала просте механічне перетворення: натискання клавіші X генерувало код Y, який перетворювався монітором клавіатури в послідовність одного або більше символічних кодів (рис. 2.3). Процес діалогу діє як пасивний посередник. Йому не потрібні знання про внутрішню структуру робочих процесів, про дані з якими він маніпулює, а необхідні лише деякі правила перетворення форматів. Адаптивні засоби, які розглядаються в п. 3.4, забезпечують лише розширення цих механізмів, які бажано застосовувати користувачеві.

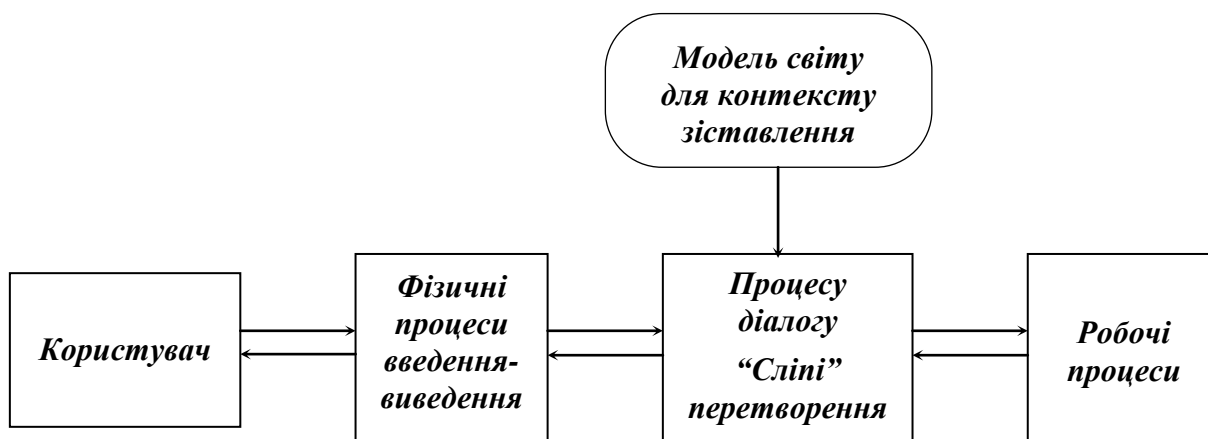
"Інтелектуальні" інтерфейси [1] відрізняються тим, що перетворення, які до цього часу виконувались через механізм визначень, повинні виконуватись в контексті предметної області, що відображається. Такий інтерфейс повинен мати деякі знання про світ задачі в якій функціонує він та користувач (рис. 2.4).



*Рис. 2.3. Неінтелектуальний інтерфейс користувача*

Необхідно зазначити, що "інтелектуальні" інтерфейси поки що не забезпечують природного діалогу людини з комп'ютером, а лише розширюють взаємодію між людиною і комп'ютером за допомогою:

- збільшення діапазону способів введення і виведення, за допомогою яких виконується взаємодія. Наприклад, за рахунок, мовного введення та виведення;
- збагачення граматики введення-виведення;
- спроби кооперації з користувачем для досягнення мети задачі.



*Рис. 2.4. Інтелектуальний інтерфейс користувача*

Важливою особливістю інтелектуальних інтерфейсів є те, що вони використовують методи теорії розпізнавання образів для інтерпретації вхідних повідомлень від користувача в світлі системної моделі світу.

Для реалізації інтелектуальних інтерфейсів, необхідний процесор великої потужності і великі об'єми пам'яті. Однак, якщо навіть припустити, що подібний процесор існує, залишаються проблеми представлення образів в системній моделі і удосконалення старих образів.

## КОНТРОЛЬНІ ПИТАННЯ

1. Які основні структури діалогу використовуються в комп'ютерних системах?
2. Охарактеризуйте відомі структури діалогу як різновидність структури типу "запитання-відповідь".
3. Які аналоги структури діалогу типу "запитання-відповідь"?
4. Переваги, недоліки, застосування структури діалогу типу "запитання-відповідь".
5. Які аналоги структури діалогу типу "меню"?
6. Які відомі формати представлення "меню" на екрані?
7. Які переваги і недоліки "меню"?
8. Наведіть основні характеристики структури діалогу типу "екранна форма".
9. В яких випадках використовується структура діалогу типу "екранна форма"?
10. Поясніть принцип роботи діалогу командного типу.
11. Які основні способи завдання параметрів в діалозі командного типу та їх переваги і недоліки?
12. Проаналізуйте діалог командного типу з точки зору користувача.
13. Приведіть схему проходження інформації в звичайних інтерфейсах.
14. Приведіть схему "інтелектуального" інтерфейсу.
15. За рахунок чого інтелектуальні інтерфейси розширюють взаємодію з користувачем?
16. Дайте порівняльний аналіз відомих структур діалогу.



## 3. ОСНОВИ ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА

### 3.1. Підтримка користувача

Структура діалогу є одним з основних факторів, який впливає на зручності роботи з ним. Однак, для успіху діалогу важливі не тільки рамки в межах яких іде обмін повідомленнями, але і зміст цих повідомлень. Можна виділити такі види підтримки користувача [1,6]:

- підказки;
- повідомлення про помилки;
- довідка;
- керівництво користувача.

Підказки є основою підтримки користувача в діалогах, що керуються комп'ютером. Якщо ця базова підтримка недостатня і користувач допускає помилки при введенні інформації, система повинна виводити повідомлення про помилки, які давали б достатню інформацію про помилку і шляхи її виправлення. Необхідно уникати загрозливих формулювань у повідомленнях про помилки. Крім того, необхідна додаткова підтримка у вигляді довідкової інформації, яка виводиться на екран по запиті користувача, а також надрукована копія "керівництва користувача" (зовнішня документація).

Довідкова інформація і повідомлення про помилки мають багато спільного. Вона повинна бути:

- інформативною;
- своєчасною (доступною в будь-якій точці діалогу);
- відповідати контексту, в якому була викликана.

Щоб відповідати цим вимогам довідкова система повинна мати ієрархічну структуру.

Зовнішня документація в діалогових системах інша, ніж в системах з пакетною обробкою, оскільки користувач не в змозі під час діалогу користуватись об'ємним керівництвом. Достатньо щоб вона містила:

- загальний огляд в межах 10 сторінок;
- опис того, як необхідно почати роботу з системою;

- довідник стандартних засобів, наприклад, в вигляді кишенькового довідника;
- довідкові написи на апаратурі;
- список несправностей.

### 3.2. Розміщення інформації на екрані

На екрані повинна бути присутня лише та інформація, яка необхідна користувачеві в даний момент. Інформація повинна бути організована так, щоб її розташування відображало логічні зв'язки, а формат був зручним для користувача. Для досягнення узгодженості, розміщення даних на всіх екранах повинно відповідати стандартному шаблону екрана (рис. 3.1). З незначними змінами цей

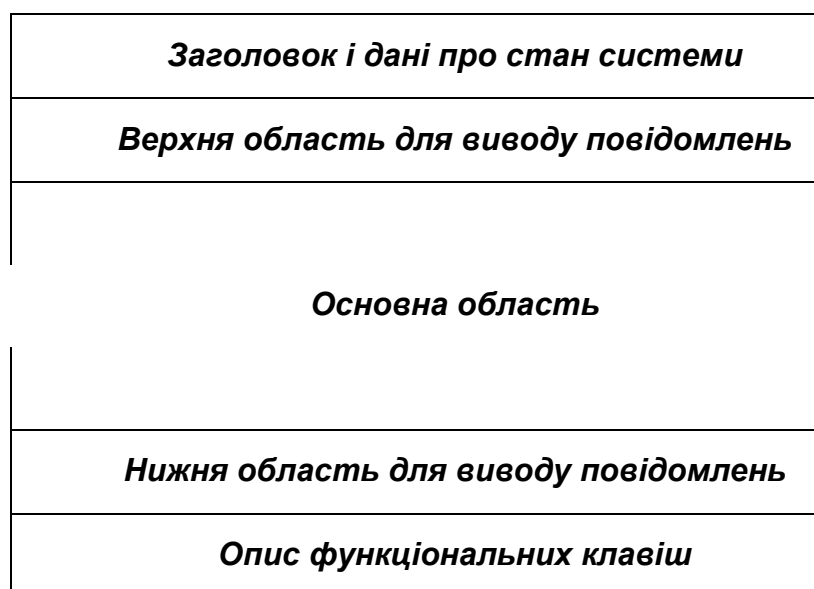


Рис. 3.1. Шаблон екрана

шаблон був прийнятий за основу при проектуванні панелей (вікон) в CUA (Common User Access) одній з чотирьох компонент стандарту корпорації IBM [5,6], яка визначає правила проектування текстових і графічних інтерфейсів користувача.

Важливе значення для сприйняття інформації має щільність її розташування на екрані. Щільно розташований текст на екрані уповільнює процес його сприйняття. Тому рекомендується додержуватись таких правил:

- залишати вільною приблизно половину екрана;

- залишати пустий рядок після кожного 5-го рядка таблиці;
- залишати 4-5 пропусків між стовпцями таблиці.

Якщо пов'язану інформацію необхідно розбити на декілька екранів, то необхідно уважно слідкувати за тим, щоб на одному екрані була вся необхідна інформація для вирішення конкретної задачі.

Виділення областей екрана притягує до них увагу користувачів і полегшує орієнтування в інформації розташованій на екрані. Найбільш часто використовується колір для виділення областей екрана. Однак і тут необхідно дотримуватись деяких правил:

- використовувати мінімальну кількість кольорів на одному екрані, не більше трьох або чотирьох;
- використовувати для виділення даних більш яскраві кольори, а більш спокійні відтінки для фону;
- для виділення двох областей - чорний колір або колір з одного кінця спектру використовується для однієї з них, а білий колір або колір з середини спектру для іншої з них;
- необхідне проведення експерименту на реальному екрані.

Оскільки розміщення даних на екрані найбільш часто виконується методом суб'єктивної експертизи, то необхідна автоматизована система розміщення даних на екрані.

### 3.3. Час відповіді системи

Час відповіді системи - це інтервал між подією і реакцією системи на неї. Що таке час відповіді системи пояснює рис. 3.2.

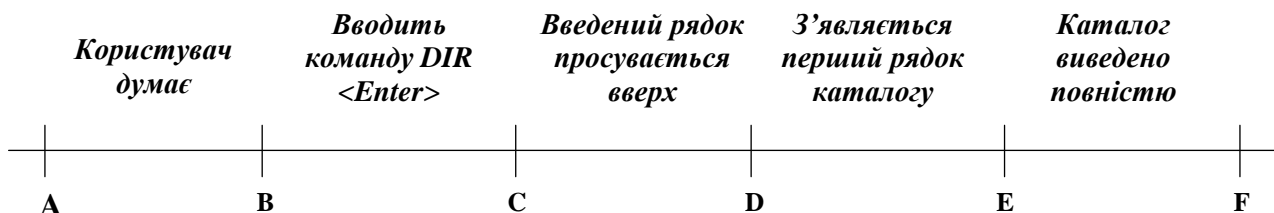


Рис. 3.2. Час відповіді на прикладі команди *Dir* - вивести каталог

Вважається, що час відповіді системи - це інтервал часу CE, тобто час від моменту введення останнього символу запиту до моменту виведення системою першого символу повідомлення, що запитувалось. Інтервал CD - називають "простою відповіддю".

Не існує єдиного прийняттого часу відповіді системи, а лише декілька видів цього часу. Час відповіді можна класифікувати таким чином [1]:

- 0,1-0,2 с.: підтвердження фізичних дій, наприклад натискання будь-якої клавіші;
- 0,5-1 с: відповідь на прості команди;
- 1,0-2,0 с.: зв'язний діалог;
- 2,0-4,0 с.: обробка складних запитів.

Якщо час відповіді більший 15-20 с., то це не діалогова система.

Людина розбиває свою діяльність на етапи, що відповідають порціям інформації, яку вона може зберігати в своїй короткочасній пам'яті. Завершення етапу називається клаузою. Тому затримки перед завершенням етапу неприпустимі, оскільки інформація в короткочасній пам'яті вимагає постійного поновлення і легко стирається. Щоб знизити негативний вплив тривалого часу відповіді, система повинна дозволяти користувачам самим встановлювати клаузи в своїх відповідях за рахунок випереджального введення відповідей та локальної обробки.

Інший спосіб зниження негативного впливу тривалого часу відповіді, це дати користувачеві впевненість, що система все ще працює. Наприклад, виводиться напис "очікуйте", або інформація про стан програми – наприклад, "оброблено 10 % записів". Часто виводяться графічні зображення в вигляді лічильників.

### **3.4. Види адаптації в діалогових системах**

Для діалогової системи дуже важливе значення має її здатність адаптуватись користувачем або самостійно до будь-якого можливого рівня підготовки оператора. Існує три види адаптації [1]:

- фіксована;
- повна;
- косметична.

При фіксованій адаптації користувач явно вибирає рівень діалогової підтримки. Очевидно, що новачку потрібно більше засобів підтримки, ніж експерту. Було сформовано правило двох рівнів, відповідно якому система повинна забезпечувати два види діалогу:

- детальний діалог, який забезпечує всебічну підтримку користувача;
- короткий діалог, який призначений для експертів і забезпечує невелику підтримку користувача або зовсім без неї.

Правило двох рівнів було розширене до правила "N" рівнів, при цьому пакет програм дозволяє користувачу явно вибрати один з "N" рівнів діалогу.

Косметична адаптація забезпечує гнучкість діалогових стилів без врахування поведінки користувача і без однозначного вибору ним конкретного стилю діалогу. Це досягається шляхом застосування:

- скорочених і частково збіжних запитів;
- синонімів;
- випереджального введення відповідей;
- відповідей по замовчуванню і макросів;
- багаторівневої допомоги.

Досвідчений користувач може застосовувати їх на свій розсуд. Вони є косметичними в тому сенсі, що як і косметика вони вносять поверхові зміни в базову структуру діалогу, але є при цьому дуже корисними, так як зменшують втому і роблять інтерфейс людина-комп'ютер більш універсальним.

При повній адаптації діалогова система прагне побудувати модель користувача, яка змінюється по мірі роботи останнього з системою і визначає стиль діалогу, адаптуючи його в залежності від цих змін. Дослідження останніх років направлені на те, як в рамках інтерфейсу людина-комп'ютер розробити і підтримувати моделі користувачів на базі методів штучного інтелекту як основи для адаптації.

### 3.5. Опис діалогу

Для представлення діалогу широко використовуються нарівні з макетами екрана мережі переходів та продукційні системи. Мережі переходів відображають динаміку діалогу, але швидко стають громіздкими, продукційні системи більш компактні, але не показують динаміку діалогу. Ні один з цих двох методів не відображає візуального виведення, тому їх можна вважати доповненням макетів екрана.

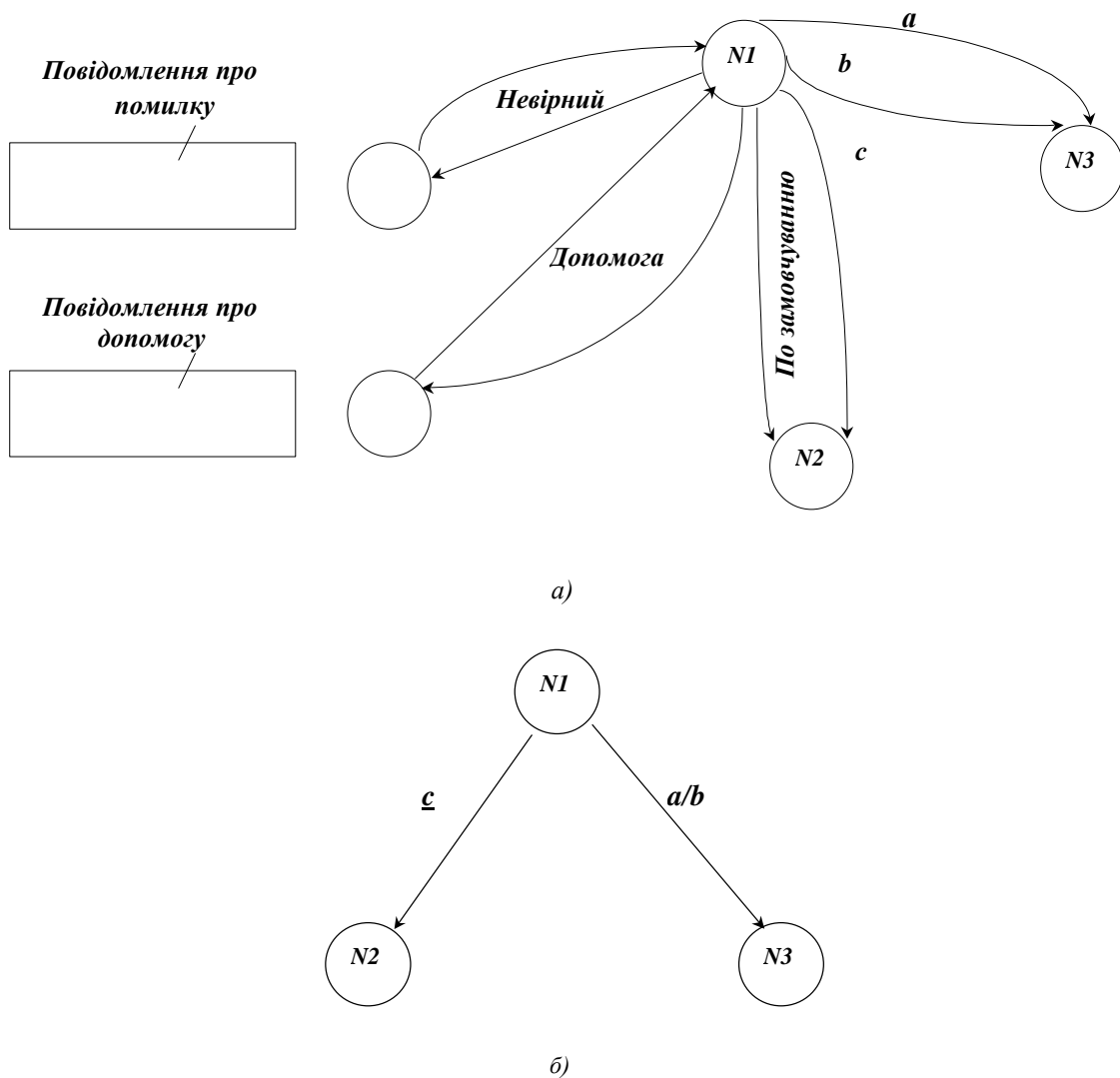


Рис 3.3. Згортка мережі переходів

Макет екрана представляє моментальний знімок діалогу в визначеному стані. Навіть, якщо макети представлені ієрархічно, вони дають погану уяву про те, як діалог переходить з стану в стан.

При опису діалогу за допомогою мережі переходів, кожний стан діалогу представляється пронумерованою вершиною графа(рис. 3.3 а). Вершина - це деяка точка в якій діалог виводить повідомлення користувачеві або вимагає вхідного повідомлення від користувача. Зв'язки між вершинами позначаються направленими дугами. Мітка на дузі визначає умову, при виконанні якої можливий перехід. Можна виділити три типи вершин в діалозі:

- вершина, в якій виводиться повідомлення користувачеві з запитом на введення. Передача на сусідню вершину залежить від контексту введеного повідомлення.
- вершина, в якій виводиться повідомлення користувачеві без запиту на введення, керування автоматично передається на сусідню вершину.
- вершина, в якій виводиться повідомлення користувачеві з запитом на введення, після якого виконується безумовний перехід на сусідню вершину.

Оскільки мережі переходів швидко стають громіздкими, то їх зображають в згорнутому вигляді (рис. 3.3 б):

- вершини, в яких виводяться допоміжні повідомлення не показуються як окремі вершини;
- якщо пара вершин зв'язується декількома дугами(показує наявність синонімів), то в згорнутому вигляді зв'язок зображується однією дугою і позначається як показано на рисунку.

Кожна вершина в мережі переходів представляє окремий стан в розвитку діалогу і утворює точку перемикання в діалозі.

Перемиканням можна керувати за допомогою структури даних (рис.3.4), яка встановлює відповідність між умовою і наступною вершиною.

Умова	Наступна вершина
C(1)	N(1)
C(2)	N(2)
...	...
C(k)	N(k)

*Рис. 3.4. Структура керування маршрутизацією в вершині*

Різні типи вершин в мережі переходів характеризуються різними структурами керування маршрутизацією в вершині. Розрізняють два способи організації програмного забезпечення в вершині мережі:

- через підпрограми;
- обробка в вершині визначається через дані - діалог, що налаштовується за таблицею.

Другий спосіб має переваги, оскільки дані змінити простіше.

Продукційні системи зручні для опису синтаксису вхідних повідомлень командної мови. Основою цього підходу є граматичний розбір команди. Граматичний розбір команди - це процес перевірки послідовності знаків на предмет відповідності їх правилам виводу. Наприклад, щоб перевірити, чи є вірною конструкція CHECK A: MYFILE.PAS система будує дерево розбору (рис. 3.5).

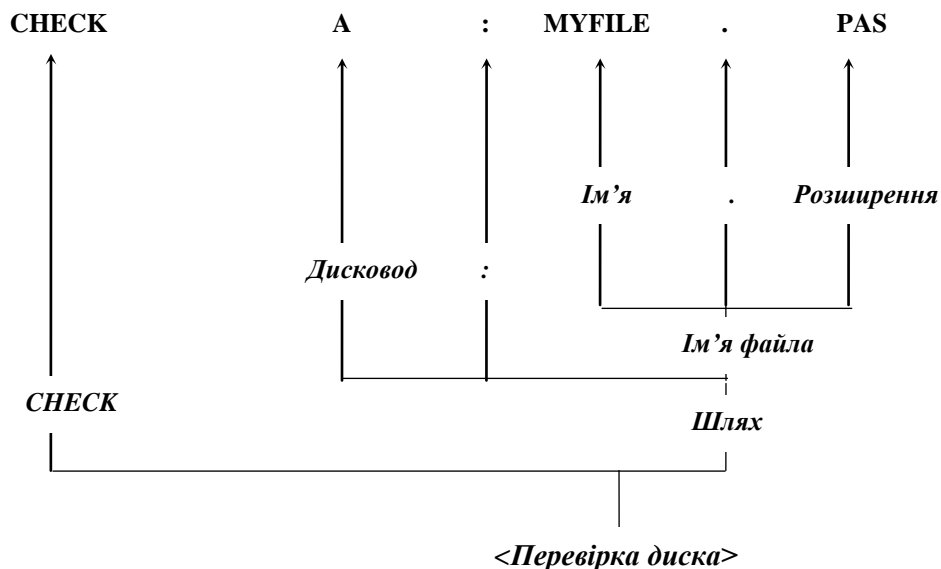


Рис. 3.5. Дерево розбору для рядка CHECK A: MYFILE.PAS

Правила виведення - це компактний спосіб визначення допустимих форматів командної мови, який знаходить також широке застосування для визначення вірних виразів в мовах програмування.



## КОНТРОЛЬНІ ПИТАННЯ

1. Які види підтримки користувача застосовуються в діалогових системах?
2. Дайте характеристику довідкової інформації.
3. Дайте характеристику зовнішньої документації.
4. Приведіть типовий шаблон екрана.
5. Яка щільність розміщення інформації на екрані?
6. Які правила виділення кольором інформації на екрані?
7. Що таке час відповіді системи?
8. Приведіть класифікацію часу відповіді системи.
9. Як знизити негативний вплив тривалого часу відповіді системи?
10. Дайте порівняльний аналіз основних видів адаптації в діалогових системах.
11. Дайте характеристику фіксованій адаптації.
12. За рахунок чого досягається косметична адаптація?
13. Дайте характеристику повної адаптації.
14. Які формальні методи застосовуються для опису діалогу?
15. Дайте порівняльний аналіз відомих методів опису діалогу.
16. Які способи організації програмного забезпечення в вершині діалогу?

## 4. БАГАТОВІКОННІ WIMP-ІНТЕРФЕЙСИ

Успіхи досягнуті в останні роки в області електронних технологій дозволили значно зменшити собівартість та підвищити потужність комп'ютерних засобів. Їх широке застосування в різних галузях господарства та побуті, привело до значних змін в інтерфейсі людина-комп'ютер. Ці зміни відносяться головним чином до взаємодії з непідготовленими або непостійними користувачами. Типовою серед цих розробок є багатовіконний WIMP-інтерфейс, який можна розшифрувати так:

W - інформація представлена користувачеві на екрані дисплея в виді декількох вікон (windows);

I - об'єкти, з якими система має діло представлені піктографічно в вигляді ікон (icons);

M - вибір об'єктів виконується маніпулятором типу "миша" (mouse);

P - означає меню, яке автоматично впливає (pop-up) на екрані або яке користувач може "витягнути" (pull down) з рядка меню, розміщеного в верхній частині екрана.

WIMP - інтерфейси підтримуються такими операційними системами як Apple Mackintosh, Widows, OS/2 та іншими [7,8,9].

### 4.1. Основні аналоги

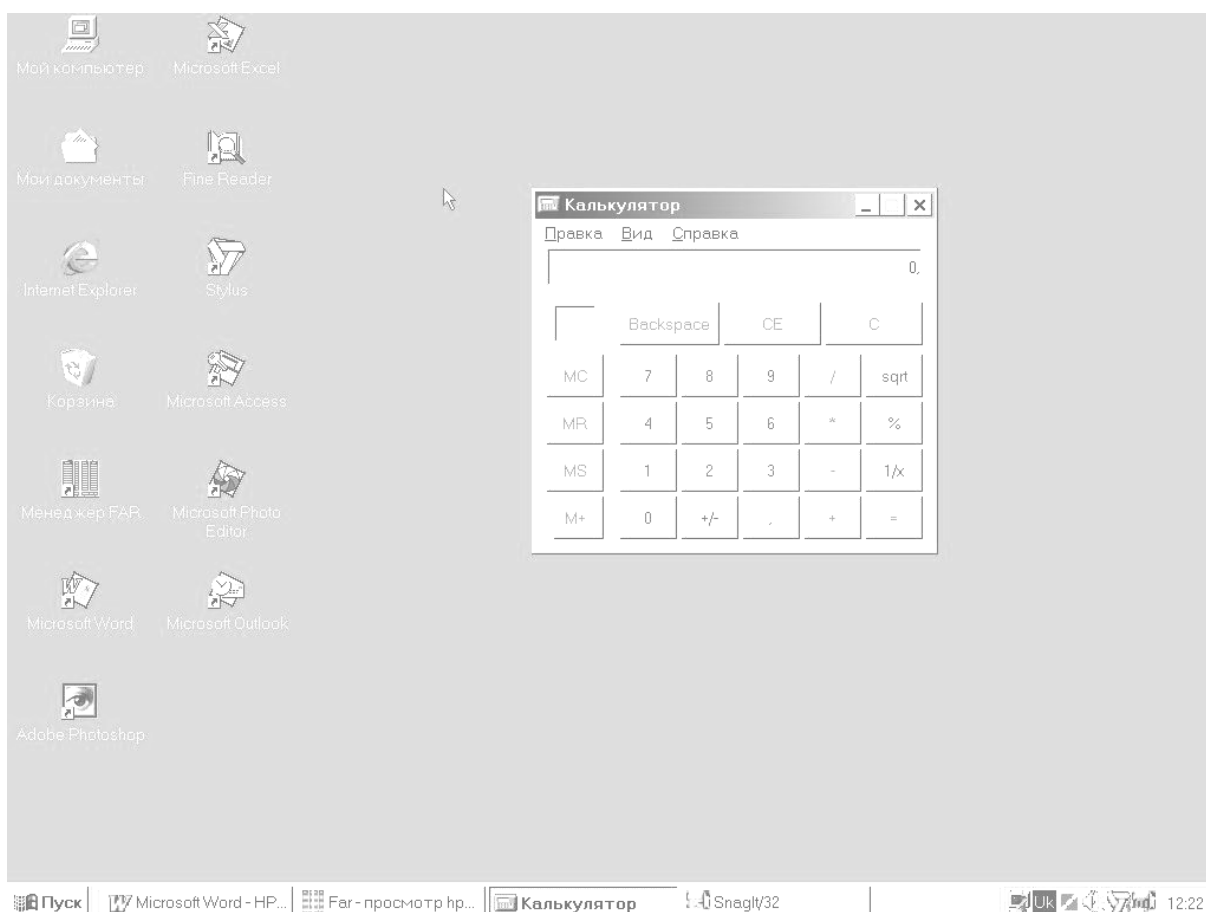
В основу WIMP-інтерфейсів покладені три основні метафори [1,5]:

- метафора "конкретний об'єкт";
- метафора "що бачиш - то і одержиш";
- метафора "робочий стіл" та "буфер вирізок".

Метафора "конкретний об'єкт" побудована за аналогією з канцелярським діловодством, де елементи задач представляються конкретними об'єктами. Файли - це пачки паперів, каталоги - висувні ящики в картотеці і т.д. Канцелярські операції передбачають фізичні дії над цими об'єктами. Аналогічно в комп'ютерних системах, наприклад, файл може бути скопійований або стертий "буксировкою" його маніпулятором типу "миша".

Метафора, що "бачиш - то і одержиш" описує такі інтерфейси, в яких фактичний ефект будь-якої дії зразу відображається на екрані дисплея. Наприклад, якщо користувач хоче віддрукувати частину тексту курсивом, то і на екрані вона повинна бути набрана курсивом.

Робочий стіл може містити документи різних форматів, які можуть відноситись до різних задач або відображати різні етапи вирішення однієї задачі (рис. 4.1.).



*Рис. 4.1. Робочий стіл ОС Windows© 98*

Людина може знайти і внести зміни в будь-який з цих документів, перекласти листки з однієї папки в іншу. Аналогічно в комп'ютерних системах метафора "робочий стіл" передбачає, що інтерфейс забезпечує користувачеві доступ до множини різних інформаційних джерел і форматів, дозволяє користувачеві легко переключатись з одного джерела на інше. Також WIMP-інтерфейси забезпечують можливість "вирізки" частини документа з копіюванням її в "буфер вирізок" і наступною вставкою в інші документи стільки копій елементів даних скільки необхідно.

## 4.2. Вікна

Вікно є спеціальна область (звичайно прямокутної форми) фізичного екрана за допомогою якої користувач оглядає окремі аспекти своєї взаємодії з задачею. На відміну від безвіконного середовища, де вихідні процеси записуються в область пам'яті зарезервовану для відображення фізичного екрана, багатовіконне середовище звичайно вводить проміжний крок як показано на рис. 4.2.

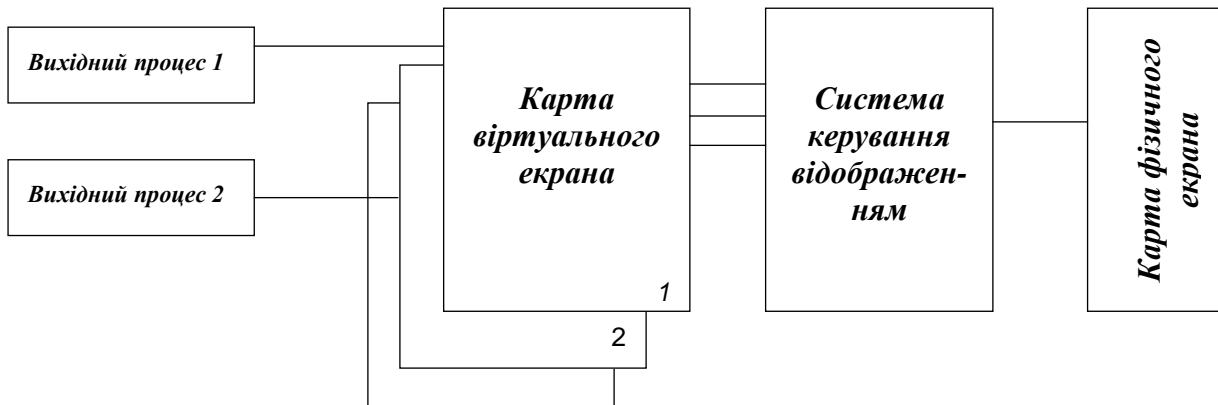


Рис. 4.2. Віртуальні екрани в багатовіконному середовищі

Вихідний процес записує інформацію в допоміжний буфер - віртуальний екран. Система може підтримувати декілька допоміжних буферів одночасно. Ці буфери можуть мати будь-який розмір і розташовуватись в будь-якій зручній ділянці пам'яті. Аналогічно як для вихідних процесів є буфери виведення, для вхідних процесів необхідні допоміжні буфери введення, оскільки в персональному комп'ютері є тільки один фізичний буфер клавіатури. Класифікація допоміжних буферів приведена на рис. 4.3.



Рис. 4.3. Класифікація допоміжних буферів

Буфер виведення - це буфер, який діалог заповнює через вихідний процес, користувач не може змінити його вміст. Буфер редагування - це буфер, вміст якого може змінювати як користувач, так і діалог. Буфер введення - це буфер, в який користувач вводить дані через вхідний процес. В багатовіконному середовищі система повинна зберігати інформацію про відкриті буфери, позицію курсору в кожному буфері та інше.

Вікно можна розглядати як відображення визначеної порції інформації з деякого буфера на фізичний екран. Кожне вікно має один буфер з яким воно зв'язане, але кожний буфер може мати зв'язок з декількома вікнами. Функції керування вікнами забезпечує система керування відображенням. Вона відповідає за:

- відкриття та закриття вікна;
- переміщення вікна відносно проміжного буфера;
- переміщення вікна відносно екрана;
- зміну розмірів та атрибутів вікна.

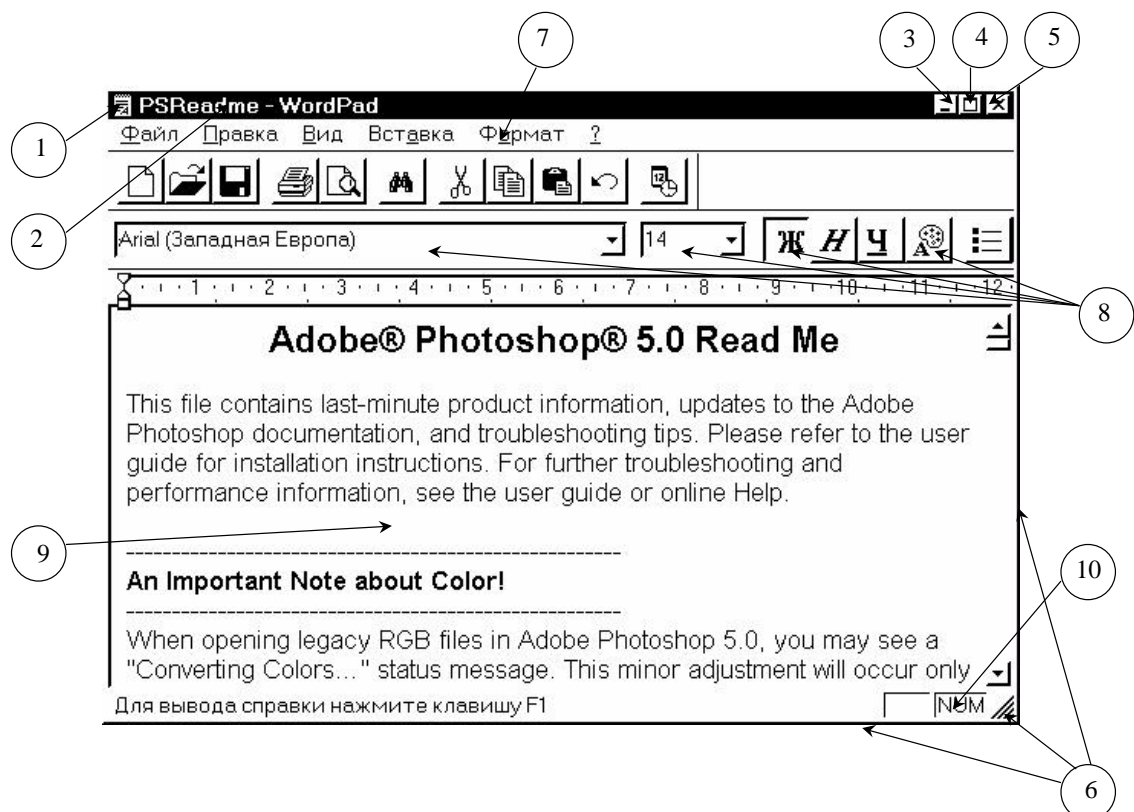


Рис. 4.4. Типове вікно Windows 95

Різноманітністю вікон є піктограми. Піктограми представляють маленьке вікно з зображенням, яке відображає вміст буфера з яким

воно зв'язане. Це вікно може розкриватись для відображення вмісту цього буфера. Піктограми - це корисний засіб нагадування, оскільки вони займають невелику площу і можуть постійно знаходитись на екрані.

Типове вікно (або панель - різниця буде пояснена в розділі 5) ОС Windows 95 приведене на рис. 4.4 [7]. Воно містить шість основних елементів:

1. Віконне меню(Control menu): містить декілька команд для налагодження вікна.
2. Область заголовка(Title bar).
3. Кнопка згортання(Minimize button):згортає вікно до кнопки на лінійці задач.
4. Кнопка відновлення / розгортання (Restore/Minimize button): дозволяє збільшити вікно до розмірів екрана або повернути його в стан, який був до розгортання.
5. Кнопка закриття(Close button): закриває вікно.
6. Краї(Edges): за допомогою миші "схвативши" будь-який край вікно можна розтягнути до необхідного розміру.

Внутрішня інформація вікна залежить від його призначення. Але є чотири спільних елементи в середині кожного вікна:

7. Рядок меню (Menu bar): список команд, який розташовується зразу під областю заголовка.
8. Панель інструментів: ряд кнопок, команд, які знаходяться під панеллю меню.
9. Робоча область (Work area).
- 10.Рядок стану (Status bar): смужка допоміжної інформації.

Крім того вікна можуть мати "лінійки прокрутки", якщо об'єм інформації буфера перевищує розмір вікна, поля введення, кнопки підтвердження дії, відміни і виходу [7,8].

### **4.3. Переваги і недоліки WIMP-інтерфейсів**

Багатовіконна технологія забезпечує користувача готовим доступом до більшого об'єму інформації в порівнянні з одним екраном. Система керування вікнами забезпечує підвищену щільність інформації на екрані. В режимі перекриття вікон

користувач може легко переміщувати інформацію з заднього плану на передній. Основні переваги багатовіконної технології такі:

1. Вікна дають доступ до множини джерел інформації. Користувач може об'єднувати інформацію з декількох джерел. Наприклад, зображення одержане за допомогою графічного пакета може бути об'єднане з текстом, одержаним за допомогою пакета підготовки тексту.

2. Споріднену інформацію користувач може досліджувати на різних рівнях деталізації. Наприклад, одне вікно може відображати графік, який характеризує тенденції в торгівлі, а інше цифри, на основі яких побудований графік.

3. Вікна і піктограми дають можливість розширити пам'ять користувача і сприяють ефективному притягненню його уваги. Частково перекрите вікно все ж нагадує користувачеві про свою інформацію; піктограми нагадують про доступні можливості.

4. Пряме маніпулювання звільняє користувача від необхідності вивчати спеціальну комп'ютерну семантику і синтаксис для взаємодії з системою.

5. Інтерфейси прямого маніпулювання характеризуються високим ступенем узгодженості. Велику різноманітність операцій можна виконати невеликим набором механізмів.

Однак, багатовіконні технології мають також ряд недоліків. Основні з них такі:

1. Метафора "робочого стола" може бути аналогією не тільки акуратного робочого стола, але і стола на якому можливий безпорядок. Невпорядкований, дезорганізований стіл може значно скоротити об'єм інформації, який отримує користувач. Це можливо, оскільки концепція вікон дозволяє користувачеві змінювати розміри та позицію вікон.

2. Оскільки користувач може керувати багатовіконним екраном, то існує небезпека, що він буде витратити більше часу на роботу з дисплеєм, а не з задачами.

3. Піктограми не завжди точно відображають зміст і вимагають великої уваги для розуміння того, що на них зображено.

4. Пряме маніпулювання не обов'язково кращий тип інтерфейсу. Наприклад, групування файлів "резиною ниткою" є менш швидким

в порівнянні з універсальними засобами, які надаються командними мовами.

5. Підтримка багатовіконного WIMP-інтерфейсу, особливо в графічних режимах, вимагає значних затрат ресурсів комп'ютера, що зменшує ресурси, виділені для вирішення прикладних задач.

#### **4.4. Критерії проектування WIMP-інтерфейсів**

1. Зміст вікна повинен утворювати логічно-зв'язану групу, а границі кожного вікна повинні бути чітко-позначені.

2. Проектувальник повинен уникати заповнення екрана множиною маленьких вікон, тому вікна повинні появлятися спочатку в відповідних позиціях і мати відповідні розміри. Система повинна встановлювати по замовчуванню початкову позицію і розмір, з врахуванням вибору користувача.

3. Компонування індивідуального вікна і екрана в цілому повинно виконуватись з урахуванням критеріїв, розглянутих в п. 3.2

4. Розробник повинен уникати складного кодування кнопок "миші", а також складних послідовностей натискання кнопок.

5. Для підготовлених користувачів інтерфейс повинен допускати можливість використання скорочень, командних мов, функціональних ключів як альтернативу керування позицією за допомогою маніпулятора типу "миша".

6. Конфігурація робочої станції повинна мати відповідну потужність для підтримки даного інтерфейсу.



## КОНТРОЛЬНІ ПИТАННЯ

1. Охарактеризуйте багатовіконні WIMP-інтерфейси.
2. На яких метафорах ґрунтуються WIMP-інтерфейси?
3. Поясніть застосування метафори "робочого столу".
4. Поясніть застосування метафори "що бачиш то і держиш".
5. Поясніть застосування метафори "конкретний об'єкт".
6. Що таке вікна та їх типи?
7. Дайте класифікацію допоміжних буферів WIMP-інтерфейсів.
8. Які елементи містить типове вікно ОС Windows 95?
9. Проаналізуйте переваги і недоліки WIMP-інтерфейсів.
10. Приведіть основні критерії проектування WIMP-інтерфейсів.

## 5. СТАНДАРТ ФІРМИ ІВМ ПО ПРОЕКТУВАННЮ

### ІНТЕРФЕЙСІВ КОРИСТУВАЧА

#### 5.1. Аналоги і принципи CUA

Основною проблемою застосування прикладних систем є проблема їх сумісності та перенесення з однієї системи на іншу. Історично склалося так, що для різних сімейств комп'ютерних систем розроблялись окремі операційні системи і засоби організації інтерфейсів користувача. Однак, широке використання персональних комп'ютерів для керування виробничими системами привело до нових поглядів на проектування цих систем і їх архітектуру. Необхідність обміну програмами, даними, ідеями користувачів різних систем примушує розробляти стандарти і відповідні програмні засоби, які б могли вирішити проблему сумісності і перенесення програм з однієї системи на іншу. Стандарт фірми IBM відомий як Systems Application Architecture (SAA) (архітектура середовища для розробки додатків), багато в чому знімає ці проблеми. SAA містить три компоненти [5,6]:

- система інтерфейсів користувача (Common User Access- CUA);
- система комунікацій (Common Communications Support - CCS);
- система програмних інтерфейсів(Common Programming Interface - CPI)[3,4].

В той час, як CPI забезпечує перенесення прикладних програмних систем, CCS забезпечує концепцію розподільної обробки даних, основною функцією CUA є забезпечення перенесення навиків і знань користувачів в роботі з системами, що не менш важливо.

Система інтерфейсів користувача(CUA) представляє сукупність правил і принципів для керівництва при конструюванні інтерфейсів користувача. В її основу покладені концепції WIMP-інтерфейсів, а саме:

- концепція "робочого стола";
- об'єктна орієнтація;
- указівка і вибір за допомогою маніпулятора типу "миша";
- використання графіки і піктограм;

- багатовіконні технології;
- кероване проектування - керування діями користувача і зменшення інформації, яку необхідно запам'ятовувати.

Проектування діалогу в CUA виконується з урахуванням ряду принципів. Деякі з них такі:

- дії користувача повинні бути оборотними. В межах діалогу дія "відміни" ( Cancel ) повертає користувача в попередній стан;
- контекст(Context) забезпечує підтримку користувача;
- дії користувача, які є потенційно руйнівними, вимагають підтвердження;
- користувач повинен мати змогу застосовувати "мишу" і клавіатуру в будь-якій точці діалогу за своїм розсудом.

Крім загальних рекомендацій CUA точно визначає ряд параметрів. Так в CUA визначені формати панелей та описані їх елементи. Ідеї закладені в CUA стали основою інтерфейсів користувача багатьох відомих операційних систем. Розробка інтерфейсу згідно з CUA складається з проектування панелей, діалогу та вікон.

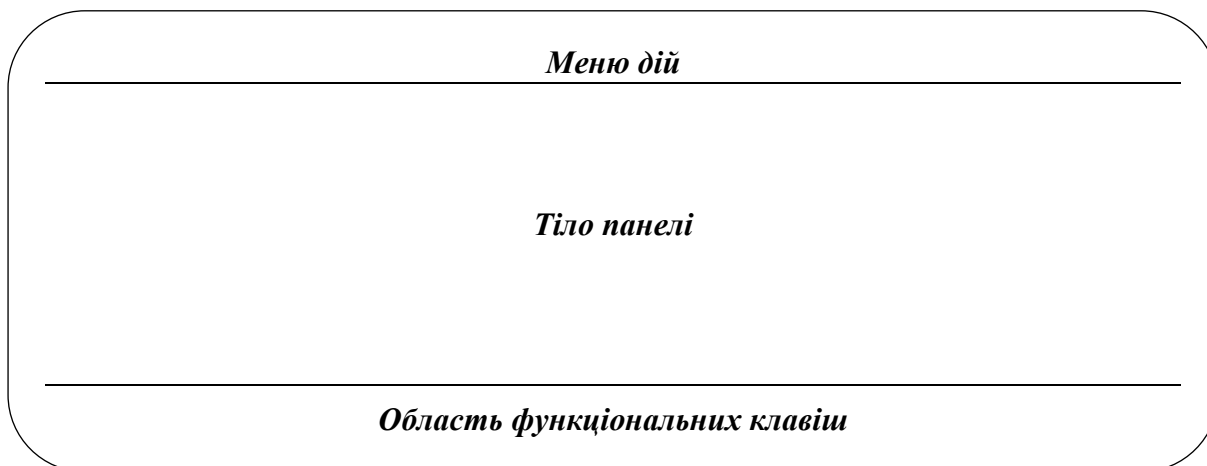
## 5.2. Панелі і вікна в CUA

В CUA визначено ряд термінів необхідних при проектуванні інтерфейсу користувача на основі багатовіконних технологій. Це такі як "ЕКРАН", "ВІКНО", "ПАНЕЛЬ", "РАДІОКНОПКА" та інші.

ЕКРАН - це поверхня дисплея, на якій розташовується інформація призначена для користувача.

ПАНЕЛЬ - це інформація, згрупована і розташована на екрані визначеним способом. Кожну панель можна представити як простір, розділений на три частини, кожна з яких містить окремий вид інформації: меню дій з меню, яке випадає, тіло панелі і область функціональних клавіш (рис.5.1).

Меню дій відображається в верхній частині екрана і надає користувачеві доступ до основних функцій додатку. При виборі дії на екрані з'являється розширення меню дій - випадне меню, яке містить список дій. Деякі панелі можуть не мати меню дій.



*Рис. 5.1. Три області панелі*

Тіло панелі може бути розділене на декілька областей. Наприклад, може містити область команд, в якій користувач вибирає команди і область повідомлень, в якій з'являються повідомлення для користувача.

Область функціональних клавiш розташована в нижній частині панелі, користувач може запросити її показ в короткій або довгій формі або виключити. Вона містить список призначень функціональних клавiш.

CUA визначає інтерфейс користувача як для текстових так і графічних додатків. Текстові додатки використовують в інтерфейсі користувача тільки текстову інформацію. Графічні додатки використовують в інтерфейсі користувача графічні елементи: радіокнопка, контактна кнопка, комірка і лінійки прокрутки (див. п. 4.2).

Комірка (Check Box): маленький квадрат за яким іде текст:

CheckBox

Використовується для кожного об'єкта поля багатозначного вибору.

Радіокнопка (Option Button): маленький кружок, за яким іде текст:

OptionButton1

Використовується для кожного об'єкта поля однозначного вибору.

Контактна кнопка: контур, деколи з округленими кутами і з текстом всередині:



Використовується для позначення безпосередніх дій.

Лінійка прокрутки: лінійка з протилежно направленими стрілками



на кінцях і повзунком на ній. Використовується, якщо об'єм інформації перевищує вміст вікна, для прокручування інформації в режимі скролінгу.

CUA визначає п'ять типів панелей (див. також п. 6.2):

- меню;
- введення;
- інформаційна;
- панель списку;
- ідентифікації додатку.

Можливо використовувати частини цих панелей для створення комбінованих панелей.

Панель меню містить один або більше списків об'єктів, з яких користувач вибирає один або декілька. Для цього панель містить поля вибору.

Панель введення відображає поля в які користувач вводить інформацію і з яких вибирає об'єкти. Є три типи панелей введення:

- панель введення параметрів відображає поля для введення значень параметрів і для вказівки режимів дій. Містить поля введення і вибору;
- панель введення з заголовками (подібно формі на папері) представляє форму документів на папері.
- панель введення з табуляцією містить поля з заголовками організовані по стовпчиках і рядках.

Інформаційна панель відображає захищену інформацію: дані, повідомлення, довідки.

Панель списку відображає список об'єктів, з яких користувач вибирає один або декілька і запитує одну або декілька дій над ними. Для кожного об'єкта одночасно може запитуватись тільки одна дія, але для різних об'єктів - різні дії.

Панель ідентифікації додатку відображає інформацію, що ідентифікує додаток і авторські права розробників.

ВІКНО - це обмежена частина екрана, в якій розташовані панелі. В CUA визначені три типи вікон: первинні, вторинні і спливальні.

Первинне вікно - це вікно, в якому починається діалог. Кожне первинне вікно може містити стільки панелей, скільки необхідно для ведення діалогу.

Вторинні вікна викликаються з первинних вікон. Це такі вікна, в яких користувач і комп'ютер ведуть діалог паралельно діалогу в первинному вікні. Наприклад, в текстовому редакторі вторинне вікно може містити панель, за допомогою якої змінюється формат документа. Вторинні вікна використовуються також для виведення довідки, яка відноситься до діалогу в первинному вікні. Користувач може переключатись з первинних вікон на вторинні і навпаки.

Спливальні вікна призначені для розширення діалогу користувача з первинним, або вторинним вікном. Одне з призначень спливальних вікон - передача різних повідомлень. Перед тим, як завершити діалог з вікном, користувач повинен завершити роботу з зв'язаним з ним спливальним вікном.

### **5.3. Уніфіковані дії діалогу**

Уніфіковані дії діалогу CUA визначає як дії діалогу, які в усіх додатках мають однакове значення. Це такі:

- відміна;
- команда;
- введення;
- вихід;
- підказ;
- регенерація;
- витягнення;
- ідентифікатори;

- клавіші;
- довідка:
  - загальна довідка;
  - зміст;
  - вказівник;
  - опис клавіш;
  - навчальна довідка.

Відміна (Cancel) це дія, яка дозволяє користувачеві вернутись на одну панель назад або з випадного меню в меню дій. Відміна повинна включатись :

- в усі випадні меню;
- в панелі, які входять в спливальні вікна, за виключенням інформаційних панелей, які містять інструкцію типу: "Нажміть введення для продовження".

Область введення команд може бути розташована на основній панелі, у вторинному та спливальному вікні. Правила використання команди такі:

- якщо область команд розміщена у вторинному або спливальному вікні, то вона появляється, коли користувач запитує уніфіковану дію діалогу "команда";
- якщо область команд знаходиться в тілі панелі, то для переходу з області команд в тіло панелі і назад використовується "команда";
- коли курсор знаходиться в полі введення команди, користувач набирає команду і запитує "введення".

Дія "введення" повідомляє додаток, що необхідно обробити інформацію панелі і завжди завершується деяким результатом. Реакція на дію "введення" змінюється від ситуації до ситуації.

Запит на вихід найбільш часто приводить до появи "спливального вікна" з такими об'єктами:

- зберегти і вийти;
- вийти без збереження;
- не виходити.

Коли користувач запитує регенерацію, то вміст поточної панелі або відновлюється в початковий стан, або регенерується для відображення поточного стану.

Витягнення запитується для відображення останньої введеної команди.

Коли користувач запитує дію "ідентифікатори", виконується ввімкнення і вимкнення відображення ідентифікаторів панелей.

Коли користувач запитує підказ, появляється спливальне вікно, яке містить інформацію, яка використовується для заповнення поля введення.

Коли користувач запитує дію "клавіші", то змінюється форма представлення області функціональних клавіш і нижньої частини випадних меню, які містять призначення клавіш. Замовчуванням для області функціональних клавіш є довга форма. Коли користувач запитує цю дію, появляється коротка форма.

Коли користувач запитує підказ, появляється спливальне вікно, яке містить інформацію, що використовується для заповнення поля введення (більш детально п. 5.4)

Коли користувач запитує довідку, появляється панель, яка містить довідкову інформацію (більш детально п.5.4).

#### **5.4. Допоміжні функції**

Важливими функціями діалогу є довідка, підказ і повідомлення. Ці функції підвищують ефективність використання додатку.

Довідка (рис. 5.2.) призначена для нагадування користувачеві синтаксису або процедури по використанню панелей або пояснення різних повідомлень. Довідка не є засобом навчання користувачів. Довідка повинна бути контекстуальною, тобто інформація, яка виводиться повинна пояснювати поточну дію діалогу. Панелі довідка можуть відноситись до панелей таких типів:

- інформаційні панелі;
- панелі меню;
- спискові панелі або панелі введення.



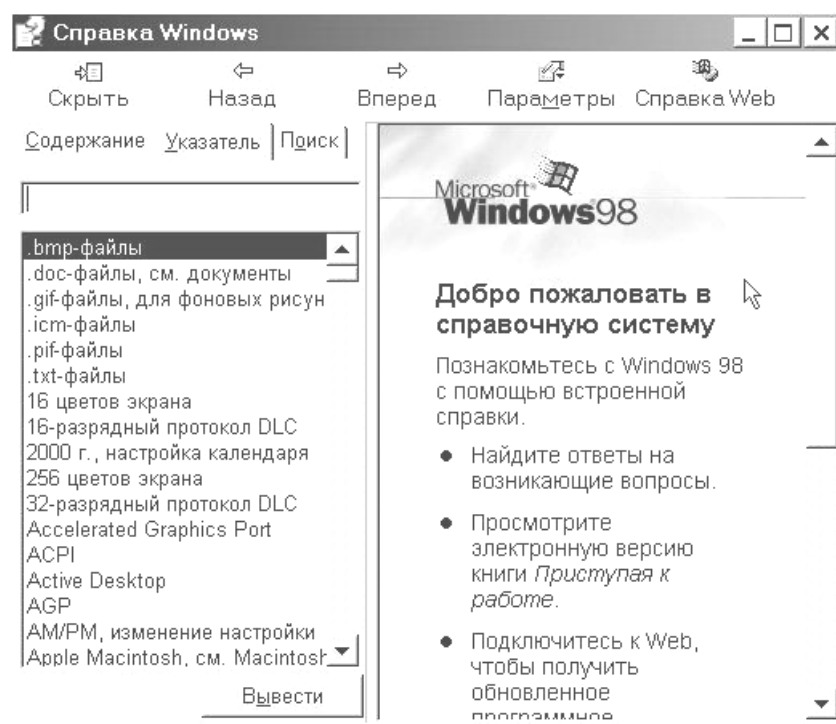


Рис. 5.2 Довідка Windows © '98

Схема розміщення панелі довідка повинна відповідати схемі вибраного типу панелі. Довідка повинна відноситись до об'єкта, на якому в поточний момент встановлено курсор, а не до місця його розташування на панелі.

"Загальна довідка" виводиться, якщо курсор не знаходиться на полі вибору або введення і містить інформацію про панель з якої її запитували.

"Опис клавіш" передбачає панель, яка описує кожен клавішу додатку і її функції. Цей перелік не повинен вказувати на доступність тієї чи іншої клавіші в поточному стані додатку. Якщо список достатньо великий і перевищує розмір вікна, то панель повинна мати лінійки прокрутки. Панель відображення клавіш звичайно є інформаційною панеллю.

"Вказівник" передбачає виведення упорядкованого за алфавітом списку тем довідкової інформації додатку. Ця панель представляється панеллю меню з лінійками прокрутки при необхідності. Коли користувач вибирає об'єкт з цього меню, з'являється панель "Довідка", яка містить інформацію з вибраної теми.

"Зміст" забезпечує виведення таблиці змісту довідкової інформації, організованої в окремі теми. Зміст може включатись у

випадальне меню "Довідка". Для цієї дії в області функціональних клавіш може бути передбачена спеціальна клавіша.

"Навчальна довідка" - це можливість додатку забезпечувати доступ до засобу навчання з панелі, з якою в даний момент працює користувач.

Вся довідкова інформація появляється в одному і тому ж вікні. Коли користувач запитує довідку, вказівник, загальну довідку, опис клавіш, зміст або навчальну довідку, то відповідна інформація заміщує панель, відображену в вікні " Довідка" в даний момент.

Деякі правила створення довідки:

- довідка повинна бути завжди доступною;
- кожна панель додатку повинна мати свою панель "довідка";
- панелі "довідка" повинні містити інформацію про поточну процедуру або про задачу, яку вирішує користувач;
- якщо додаток дозволяє вводити команди, то якщо користувач ввів команду і запросив "довідку", при умові, що курсор знаходиться в області команд, повинна появитись панель, яка описує набрану команду. Якщо користувач запитує довідку без набору команди, то повинна появитись інформація про команди загального характеру;
- уніфіковані дії діалогу і функціональні клавіші повинні діяти таким же чином, як в інших місцях додатку;
- коли користувач входить або виходить з панелі "довідка" дані, введені користувачем до запиту довідка не повинні знищуватись.

"Підказ" підказує користувачам те, як завершити роботу з даним полем введення. Коли запитується "підказ" , появляється спливальне "вікно" , яке містить панель меню з полем однозначного або багатозначного вибору. Коли користувач вибирає один або більше об'єктів з цього поля, спливальне вікно зникає. Текст вибраного об'єкта копіюється в поле введення. Тобто, завдяки підказу, користувачеві нема необхідності запам'ятовувати і набирати великі об'єми інформації. Достатньо лише вибрати необхідне.

"Повідомлення" забезпечують користувача інформацією, яку він явно не запитував, але яка йому необхідна. Наприклад, повідомлення типу:

**"Ви впевнені..."**

запитує у користувача підтвердження намірів.

У відповідності з характером інформації, яка виводиться, визначені три типи повідомлень:

1. Інформаційне повідомлення - забезпечує зворотний зв'язок про стан системи. Відповідь користувача не потрібна. Інформаційне повідомлення вказує, що функція виконується або виконалась нормально і використовується в таких випадках:

- користувач багатозадачної системи виконує фонове завдання і воно сповіщає користувача про завершення роботи;
- з'ясування поточного стану системи на основі іншої відображеної інформації може зайняти у користувача декілька секунд;
- користувачеві не ясно, чи обробка продовжується, чи завершилась.

2. "Попереджувальне повідомлення" - звертає увагу користувача на стан, який вимагає деякої дії. Користувач може відповісти на це повідомлення виконанням деякої дії, відмінної від уніфікованої дії діалогу "Відміна". Однак, виконання цієї дії не обов'язково.

3. "Критичні повідомлення" - вказують на умову, коли для продовження роботи користувач повинен виконати конкретну дію. В відповідь на таке повідомлення користувач повинен виконати конкретну дію.

Повідомлення рекомендується виводити в спливальних вікнах. Воно може видаватись в:

- інформаційній панелі;
- панелі меню;
- панелі введення.

Область повідомлення може займати будь-яку кількість рядків. При виводі попереджувальних і критичних повідомлень повинна бути передбачена звукова сигналізація.

## КОНТРОЛЬНІ ПИТАННЯ

1. На яких аналогах ґрунтується CUA?
2. Принципи проектування діалогу згідно з CUA.
3. Поясніть такі терміни як "екран", "вікно", "панель".
4. Приведіть стандартний формат панелі CUA.
5. Чим відрізняється інтерфейс користувача текстових і графічних додатків згідно з CUA?
6. Приведіть графічне представлення та призначення радіокнопки, контактної кнопки, комірки, лінійки прокрутки.
7. Які типи панелей визначені в CUA?
8. Які типи вікон визначені в CUA?
9. Які уніфіковані дії діалогу визначені в CUA?
10. Приведіть допоміжні функції діалогу визначені в CUA.
11. Приведіть правила створення довідки згідно з CUA.
12. Які види повідомлень визначені в CUA?
13. Які види довідок визначені в CUA?
14. Які дії діалогу в CUA, коли запитується "підказка"?

## 6. ЛАБОРАТОРНИЙ ПРАКТИКУМ

Метою даного лабораторного практикуму є вивчення особливостей проектування інтерфейсу користувача з використанням структур діалогу різних типів і програмних засобів різного рівня. Лабораторна робота розробляється як цілісне програмне середовище, яке у відповідності з концепціями СUA, підтримує інтерфейс користувача з різними структурами діалогу шляхом відкриття відповідних вікон. Робота повинна бути представлена у двох варіантах:

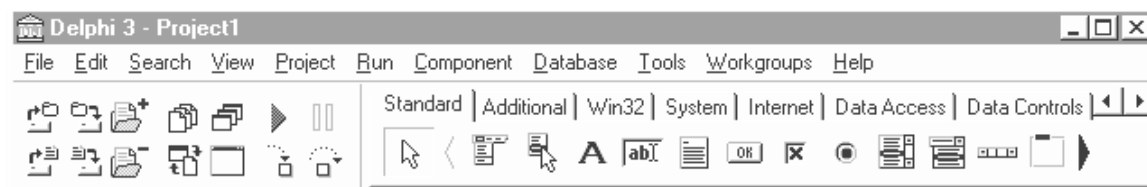
- виконана на мові С++ або Turbo Pascal;
- виконана з використання середовища візуального програмування DELPHI 3 (Object Pascal) або Borland С++ Builder 3.

Різниця між цими двома реалізаціями полягає в тому, що у першому випадку студенти вивчають основи проектування інтерфейсу користувача на більш низькому рівні близькому до системного, а використовуючи DELPHI на рівні прикладних програмістів. DELPHI містить автоматизовану систему проектування інтерфейсу користувача, яка звільняє прикладних програмістів від необхідності розробки програмних засобів підтримки інтерфейсу користувача. Вона дозволяє за допомогою простих операцій указання і вибору (наприклад, за допомогою маніпулятора типу “миша”) спроектувати необхідний інтерфейс користувача.

### 6.1. Проектування інтерфейсу користувача у середовищі Borland Delphi.

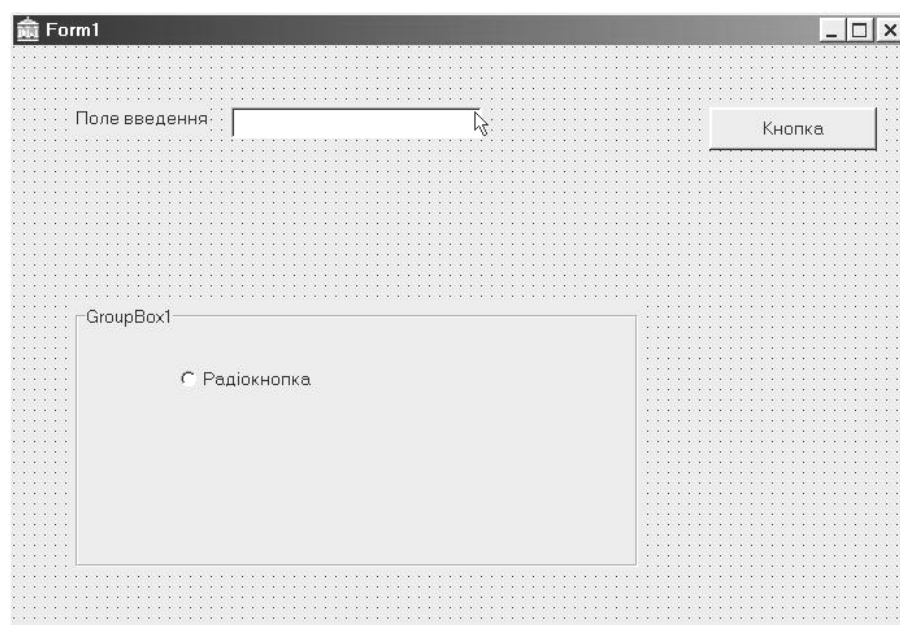
Панелі інструментів Borland Delphi 3 приведені на рис. 6.6.

В ході виконання лабораторної роботи будемо використовувати основні компоненти Borland Delphi для створення діалогу. До таких компонент можна віднести: TButton, TEdit, TLabel, TMenu, TListBox, TComboBox, TCheckBox, TRadioButton та інші. Розглянемо роботу з ними детальніше.



*Рис. 6.6. Панелі інструментів Borland Delphi 3*

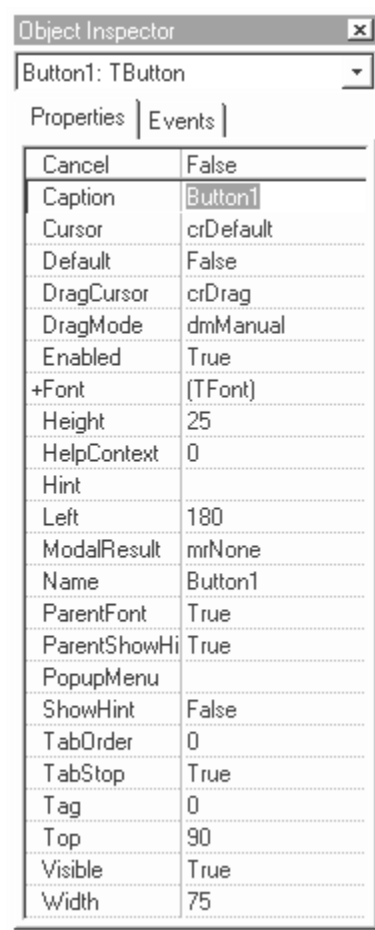
Компонент TButton являє собою звичайну стандартну кнопку (рис. 6.7). TButton - знаходиться на сторінці "Standard" палітри компонентів (рис. 6.6). Для того щоб помістити компонент на форму (рис. 6.7) виконуємо такі дії: клацаємо мишею по компоненті TButton на палітрі компонент, а потім клацаємо по формі. Для зміни геометричних розмірів отриманої кнопки можна використати Object Inspector (рис. 6.8), де слід поміняти величини Left (X-координата лівого краю кнопки), Top (Y-координата верхнього краю кнопки), Width (ширина кнопки), Height (висота кнопки). Зміну геометричних параметрів будь-якого об'єкта можна також виконати за допомогою маніпулятора типу "миша". Для того щоб перемістити об'єкт, слід просто перетягнути його "мишею" на нове місце. Для того щоб



*Рис. 6.7. Стандартна форма Borland Delphi 3*

змінити розміри об'єкта слід підвести курсор миші до границі об'єкту (при цьому курсор миші змінить свою форму) і перетягнути границю в нове місце.

Зміна напису на кнопці здійснюється шляхом зміни властивості Caption, яку теж можна поміняти у вікні Object Inspector. Якщо вікно Object Inspector в даний час відсутнє на екрані, то слід клацнути по об'єкту для якого визивається інспектор і натиснути клавішу F11.



*Рис. 6.8. Інспектор об'єктів Borland Delphi 3*

Колір кнопки (як і колір будь-якого іншого об'єкта) встановлюється за допомогою властивості Color. Один із стандартних кольорів можна вибрати у списку, що відкривається або в полі введення ввести код кольору у цифровому вигляді.

Для зміни шрифту, яким виводиться напис слід скористатися властивістю Font. Для цього слід двічі клацнути "мишею" по значку і в запиті, що відкривається вибрати характеристики потрібного шрифту.

Для того щоб написати процедуру обробки натискання на кнопку слід по ній клацнути двічі, після чого з'явиться вікно редактора коду, де і слід вписати текст процедури (між begin та end).

Всі властивості об'єктів можна задавати не лише під час розробки форми, а й міняти під час виконання програми. Скажімо, необхідно зробити так, щоб кнопка на момент запуску програми була недоступна, а потім після настання деякої події ставала доступною. Для цього слід поміняти значення властивості Enabled:

### **Button1.Enabled:=true**

Серед інших властивостей, характерних для всіх компонентів, що будуть розглянуті далі можна виділити такі: Cursor, Hint, ShowHint, PopupMenu (рис. 6.8).

Властивість Cursor задає вигляд, який приймає курсор миші при переміщенні ним над об'єктом.

Властивість Hint задає підказку, яка з'являється через кілька секунд після наведення курсору миші на об'єкт. Підказка буде видаватися лише у випадку якщо властивість ShowHint має значення true.

Властивість PopupMenu задає контекстне меню, яке з'являється при натисненні на праву кнопку миші.

Компонент TEdit знаходиться на сторінці "Standard" палітри компонентів і фактично являє собою поле введення (рис. 6.7). Основні властивості компоненти:

- Color - колір компоненти.
- Left, Right, Width, Height - задають геометричні характеристики об'єкта.
- Font вказує шрифт яким буде відображатися текст, що вводиться.
- Text це власне і є введений текст.

Властивість Text має тип string. Якщо в програмі необхідно ввести число, то для перетворення типів слід користуватися функціями strtoint (для перетворення в тип integer) , strtfloat (для перетворення в тип real)

Інші компоненти сторінки "Standard" необхідні при проектуванні інтерфейсу користувача наведені нижче.



**TMainMenu** дозволяє помістити головне меню в програму. При поміщенні TMainMenu на форму воно виглядає , як проста іконка. Іконки даного типу називають "невидимими компонентами", оскільки вони невидимі під час виконання



програми. Створення меню включає три кроки: (1) поміщення TMainMenu на форму, (2) виклик Дизайнера Меню (для цього слід двічі клацнути по іконці компоненти), (3) визначення пунктів меню в Дизайнері Меню.



**TPopupMenu** дозволяє створювати спливальні вікна меню. Меню такого типу з'являється якщо клацнути правою кнопкою "миші" по об'єкту, з яким це меню пов'язане. У всіх видимих об'єктів є властивість PopupMenu, де і слід вказати потрібне меню. Створюється PopupMenu аналогічно головному меню.



**TLabel** служить для відображення тексту на екрані. Шрифт та колір мітки задаються властивістю Font. Крім того текст, колір, шрифт можна міняти і в ході виконання програми, написавши для цього всього лише один рядок коду.



**TMemo** - інша форма TEdit. Цей компонент використовується для роботи з великими за розміром текстами. TMemo може переносити слова, зберігати в Clipboard фрагменти тексту і відновлювати їх, та інші основні функції редактора. TMemo має обмеження на розмір тексту до 32Кб, це складає 10-20 сторінок.



**TCheckBox** відображає рядок тексту з маленьким вікном біля нього. В вікні можна поставити відмітку, яка означає, що щось вибрано. Наприклад, якщо подивитись вікно діалогу налаштувань компілятора (пункт меню Options | Project, сторінка Compiler), то можна побачити, що воно складається в основному з елементів CheckBox.



**TRadioButton** (радіокнопка) дозволяє обрати лише одну опцію із декількох (рис. 6.7).



**TListBox** використовується для відображення списку з скролінгом.



**TComboBox** багато в чому нагадує ListBox, за виключенням того, що є можливість вводити інформацію в маленькому полі введення.



**TPanel** - керуючий елемент, який використовується в декоративних цілях. Для використання TPanel, слід просто розмістити компонент на формі, після чого потрібно розмістити інші компоненти на ній. Тепер при зміщенні TPanel будуть зміщуватись і компоненти розміщені на панелі. TPanel використовується також для створення панелі інструментів і вікна стану.

На сторінці Standard представлені керуючі елементи, які з'явилися ще в Windows 3.0. На сторінці Additional розміщені об'єкти, які дозволяють створити більш зручний інтерфейс користувача. Список деяких компонент, розміщених на сторінці Additional такий:



**TBitBtn** - кнопка схожа на TButton, але на неї можна помістити картинку (glyph). TBitBtn має декілька заздалегідь визначених типів (bkClose, bkOK та ін.), при виборі яких кнопка приймає відповідний вигляд.



**TSpeedButton** - кнопка для створення панелі швидкого доступу до команд (SpeedBar). Як правило, на таку кнопку поміщається тільки картинка (glyph).



**TMaskEdit** - аналог TEdit, але з можливістю форматowanego вводу. Формат задається властивістю EditMask. В редакторі властивостей для EditMask є заготовки деяких форматів: дати, валюти і т.п.



**TStringGrid** - призначений для представлення текстових даних у вигляді таблиці. Доступ до елементів таблиці відбувається через властивість Cell.



**TDrawGrid** - призначений для представлення будь-яких даних у вигляді таблиці. Доступ до елементів таблиці відбувається через властивість CellRect.



**TImage** – відображає графічне зображення на формі. Сприймає формати BMP, ICO, WMF. Якщо зображення підключити під час розробки програми, то воно запишеться в EXE файл в вигляді ресурсу.



**TShape** - призначений для відображення простих графічних об'єктів на формі: круг, квадрат і т.п.



**TBevel** - елемент для рельєфного оформлення інтерфейсу.



**THeader** - елемент оформлення для створення заголовків з змінюваними розмірами для таблиць.

Якщо потрібна додаткова інформація стосовно деякого об'єкта, то слід обрати на Палітрі об'єкт і натиснути комбінацію клавіш Ctrl-F1 – з'явиться довідник з повним описанням даного об'єкта.

Для написання будь-якої програми в середовищі Delphi недостатньо мати один файл (з розширенням pas) як було в середовищі Turbo Pascal. Необхідно мати чотири файли, які всі разом називаються проектом. Ось ці файли :

- Головний файл проекту (розширення dpr). Створюється автоматично при виборі пункту меню New Project.
- Перший модуль програми /unit/, який автоматично з'являється при створенні нового проекту. Файл називається UNIT1.PAS , але його можна перейменувати, давши йому будь-яке інше ім'я (розширення має бути pas). В процесі написання програми до проекту можуть додаватися нові модулі, а непотрібні вилучатися.
- Файл основної форми (розширення dfm), використовується для зберігання інформації про саму форму, а також про розміщені на ній компоненти. Крім основної форми до проекту можуть бути додані і інші форми (пункт меню New Form).
- Файл із розширенням .RES містить іконку для проекту, створюється автоматично і про його існування програміст може і не підозрювати.

Після компіляції програми з'являються файли з розширеннями:

- DCU - відкомпільовані модулі
- EXE – файл готовий до виконання
- ~PA, ~DP, ~DF – backup-файли редактора.

## 6.2. Методичні вказівки

Першою після запуску додатка на екран виводиться панель ідентифікації додатка. Панель ідентифікації додатка містить відомості про права власності і авторські права. Зміст її такий:

- умовне позначення за яким слідує назва системи;
- назва і версія додатка;
- реєстраційний номер додатка, якщо такий є;
- відомості про авторське право (рис.6.1)

**ПМОС**

**Найменування системи**

Назва додатку  
Версія N.NN

**Порядковий номер ZZZZ**

**Авторське право кафедри ПМОС**

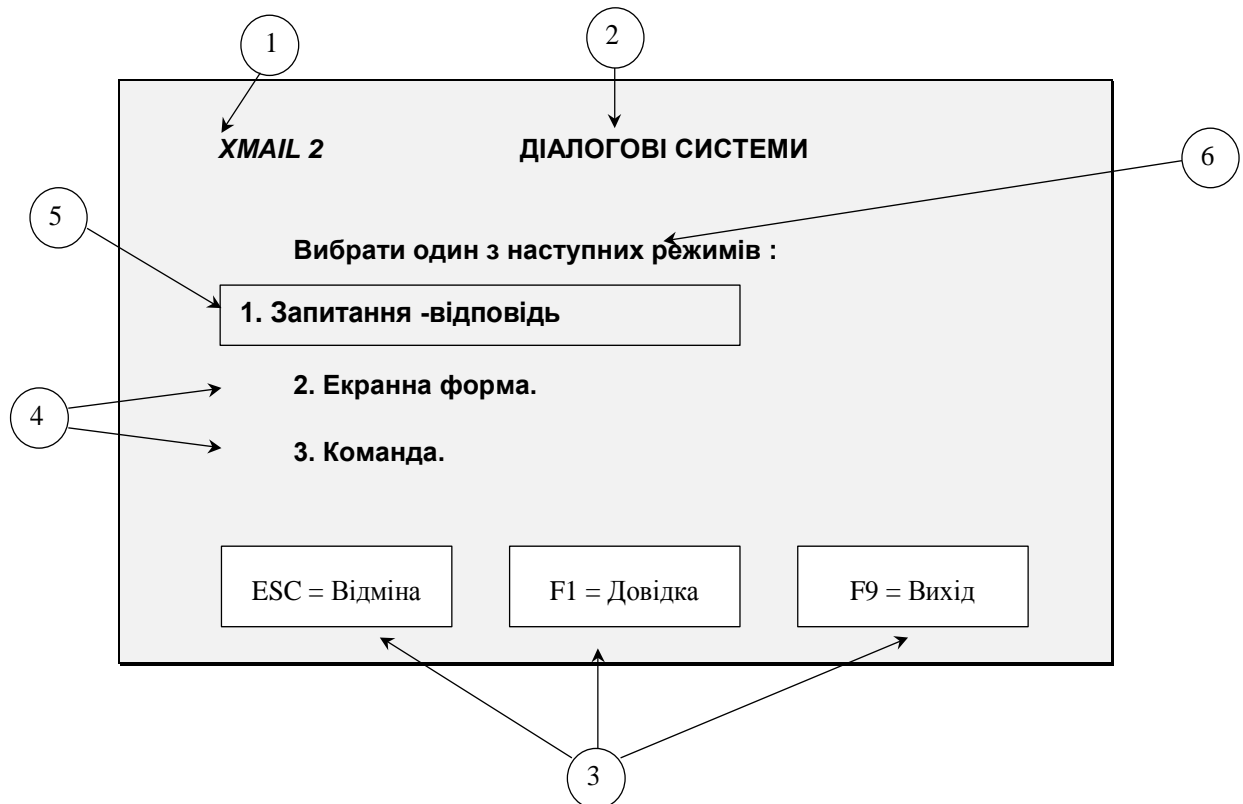
Введення

ESC=Відміна

*Рис. 6.1. Панель ідентифікації додатка*

Після натискання клавіші "введення" (або вибору за допомогою маніпулятора типу "миша") на екрані повинно з'явитись головне меню системи. Згідно з СУА панель меню містить такі елементи:

1. Ідентифікатор панелі;
2. Заголовок панелі;
3. Область функціональних клавіш (або кнопок при керуванні "мишею");
4. Об'єкт поля вибору;
5. Поле вибору;
6. Інструкцію (рис. 6.2).



*Рис. 6.2. Панель меню*

Крім приведених елементів панель меню може мати при необхідності рядок "меню дій", лінійки протяжки. Для вибору відповідного пункту меню необхідно виділену смужку помістити на нього за допомогою клавіш керування курсором або маніпулятора типу "миша" і натиснути клавішу "Введення" (Enter) або виконати вибір за допомогою "миші". В результаті розкриється нове вікно, в яке поміщається нова панель, наприклад, панель введення. Типова панель введення в CUA містить:

1. Меню дій;
2. Заголовок панелі;
3. Інструкцію;
4. Поле введення;
5. Лінійки протяжки (при необхідності);
6. Поле вибору;
7. Пояснення до поля вибору;
8. Заголовок до поля вибору (рис.6.3 ).

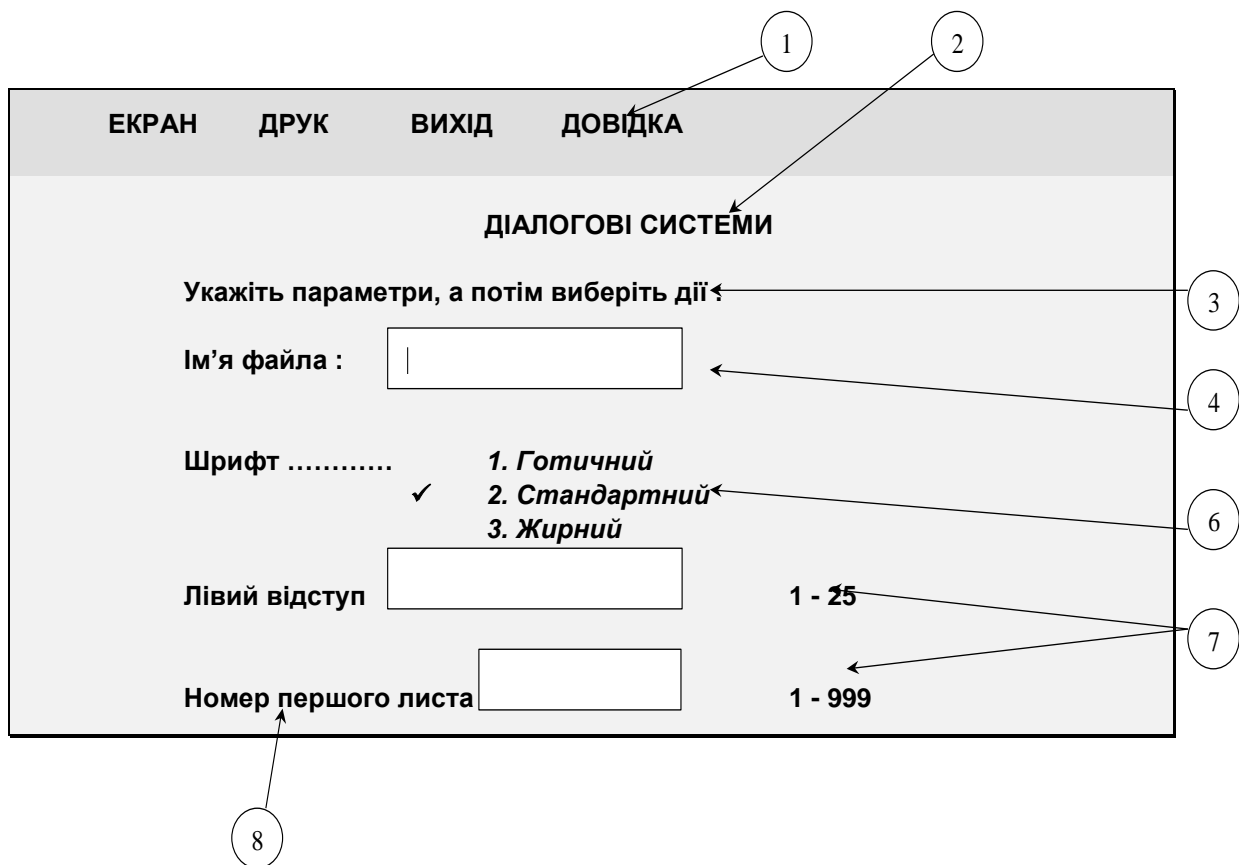


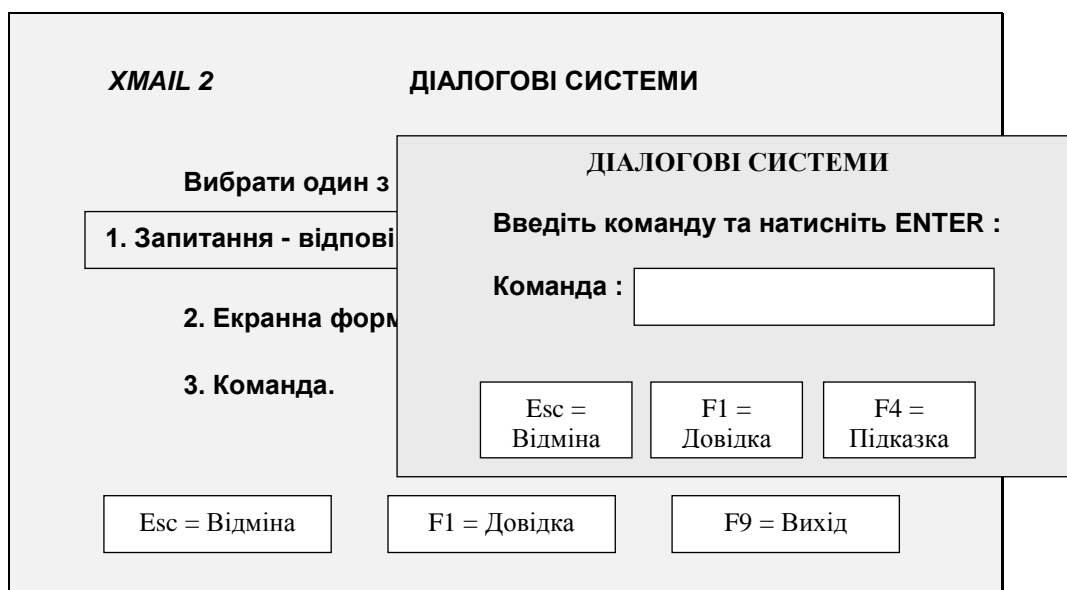
Рис. 6.3. Панель введення

Панель введення можна використовувати для реалізації структури діалогу типу "екранна форма". Для реалізації діалогу типу "запитання -відповідь" у навчальній задачі можна також використати цю панель. При досягненні нижнього краю панелі необхідно забезпечити протягання екрана вгору. Після відповіді на останнє запитання необхідно вивести відповідну форму на екран і забезпечити її запис в файл або виведення на друк за допомогою меню дій, відповідні пункти якого повинні мати можливість розкриватись з відображенням "випадного" меню.

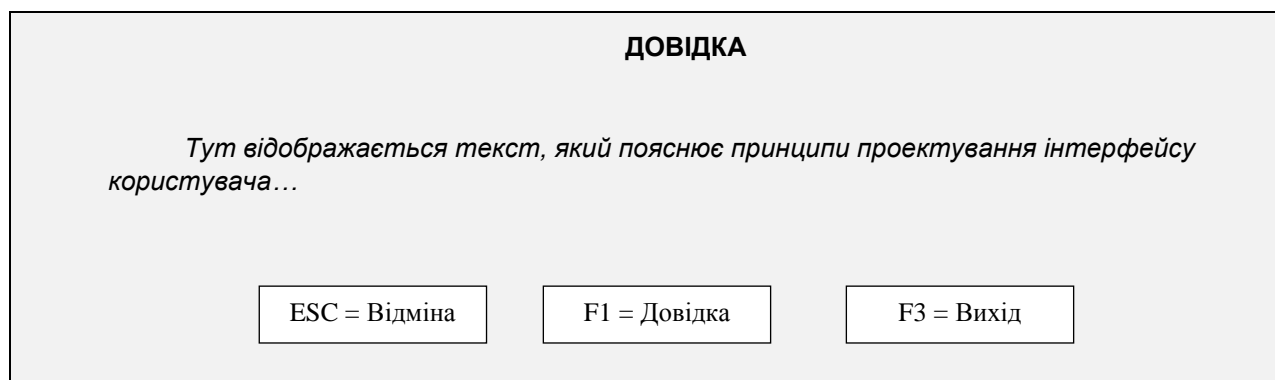
При виборі пункту меню "команда" (рис. 6.2.) на екрані повинно з'являтися (впливати) невелике вікно для введення команди (рис.6.4).

Необхідно передбачити в цій навчальній роботі можливість введення і виконання 4-5 команд.

Інформаційна панель може викликатись з будь-якої точки діалогу. Вона відображає захищену інформацію - повідомлення, довідки (рис. 6.5).



*Рис. 6.4. Вікно для введення команд*



*Рис 6.5. Приклад інформаційної панелі*

Всі панелі, крім описаних елементів, можуть містити і інші, наприклад лінійки протяжки, кнопки і т.д. Допускається застосування комбінованих панелей, які містять елементи панелей декількох типів.

### 6.3. Завдання

1. Розробити програмне забезпечення підтримки інтерфейсу користувача з використанням діалогу типу меню, команд, екранних форм:

## Приклади варіантів завдань

1.1. Розробити інтерфейс користувача інформаційної системи визначення професійної орієнтації для працевлаштування. Забезпечити можливості отримання "твердої" копії анкети, запису її на диск та інші необхідні функції.

1.2. Розробити інтерфейс користувача інформаційної системи для знімання копій паспортів. Забезпечити можливості друку копії паспорта, запису її на диск та інші необхідні функції.

1.3. Розробити інтерфейс користувача інформаційної системи для знімання копій свідоцтва про народження. Забезпечити можливості друку копії свідоцтва, запису її на диск та інші необхідні функції.

1.4. Розробити інтерфейс користувача інформаційної системи для знімання копій свідоцтва про одруження. Забезпечити можливості друку копії свідоцтва, запису її на диск та інші необхідні функції.

1.5. Розробити інтерфейс користувача системи установки параметрів дисплея. Забезпечити можливості друку протоколів діалогу , запису їх на диск та інші необхідні функції.

## 6.4. Етапи виконання роботи

**Етап 1.** Розробити проект системи і сформувати головне меню (див. рис. 6.2.) у відповідності з варіантом завдання - 6 год.

### Контрольні питання

1. Критерії розробки "меню".
2. Які формати використовуються для виводу "меню" на екран?
3. Охарактеризуйте "меню" як різновидність діалогу типу Q&A.
4. Що таке приховане "меню"?
5. Які користувачі найбільш часто використовують діалог типу "меню"?
6. Складові частини інтерфейсу людина-комп'ютер.

**Етап 2.** Розробити програмне забезпечення підтримки діалогу типу запитання-відповідь (використовуючи панель приведену на рис.6.4.) - 6 год.



### Контрольні питання

1. Критерії оцінки якості діалогу.
2. Коли найбільш раціонально застосовувати діалог типу Q&A?
3. Дайте визначення діалогу типу Q&A.
4. Хто управляє діалогом в структурі Q&A?
5. Коли використовують випереджальне введення?
6. Що таке режим ехо-друку?
7. Які види підтримки користувача застосовуються в діалозі типу Q&A.

**Етап 3.** Розробити програмне забезпечення підтримки діалогу типу "екранна форма", використовуючи панель приведену на рис.6.3. - 4 год.

### Контрольні питання

1. Які етапи розміщення інформації на екрані?
2. В якому вигляді інформація повинна виводитись на екран?
3. Як розміщується інформація на екрані?
4. Де розміщують інформацію про помилки?
5. Що і як виділяють на екрані?
6. Які якості діалогу на основі екранних форм?

**Етап 4.** Розробити програмне забезпечення панелі введення команд (див. рис.6.4.), забезпечити необхідні засоби адаптації - 6 год.

### Контрольні питання

1. Які користувачі можуть ефективно використовувати діалог командного типу?
2. Які критерії розробки діалогу командного типу?
3. Як пов'язані між собою діалог командного типу і діалог типу запитання-відповідь (Q&A).
4. Що включає в себе команда?
5. Які ви знаєте види адаптації?
6. Недоліки фіксованої адаптації.
7. В чому полягає суть методів косметичної адаптації?
8. Чи можлива повна адаптація?
9. Як організована багаторівнева допомога?

**Етап 5.** Розробити програмне забезпечення для підтримки функцій "випадних меню" (витягуються з "меню дій" - рис. 6.4.). Підготувати звіт по роботі в цілому і комплект необхідної зовнішньої документації – 6 год.

### **Контрольні питання**

1. Основні концепції і принципи CUA.
2. Дати визначення WIMP-інтерфейсу.
3. Дати класифікацію буферів WIMP-інтерфейсу.
4. Дати класифікацію панелей CUA.
5. Подати перелік зовнішньої документації в діалогових системах.
6. Дати перелік уніфікованих дій діалогу в CUA.

Примітка. При запуску програми обов'язкове виведення панелі ідентифікації додатків, інформаційна панель виводиться по запиту користувача в будь-якій точці діалогу з контекстною довідкою (рис.6.1., рис. 6.5.).

Зміст звіту:

- комплект зовнішньої документації на систему;
- опис діалогів за допомогою мереж переходів, макетів екрана та правил введення;
- текст програми.

## ЛІТЕРАТУРА

1. Коутс Р., Влейминк И. Интерфейс "Человек - компьютер": Пер. с англ. - М.: Мир, 1990.
2. Деннинг В., Эссинг Г., Маас С. Диалоговые системы "человек-ЭВМ". Адаптация к требованиям пользователя: Пер. с англ. - М.: Мир, 1984. - 112с.
3. Диалоговые системы. Современное состояние и перспективы развития/ Довгялло А.М., Брановицкий В.И., Вершинин К.П. и др. Киев: Наук. думка, 1987. - 248 с.
4. Артемьев В.И., Строганов В.Ю. Организация диалога в САПР. М.: Высш. шк., 1990. - 158 с.
5. Архитектура среды для разработки приложений. Киев: "Крещатик", 1992. - 240 с.
6. Проектирование пользовательского интерфейса на персональных компьютерах. Стандарт фирмы IBM/Под ред. Дадашова М. - Вильнюс: DBS LTD., 1992, 180 с.
7. Гукин Дэн. Реальная жизнь Windows 95/Пер. с англ. - К.: НИПФ - "ДиаСофт ЛТД", 1996. - 544 с.
8. Мюллер Дж., Нортон П. Полное руководство по Windows 95 Питера Нортон/ Пер. с англ. - М.: "Бином", 1998. - 778 с.
9. Кренц Дж., Майзел Э. Уильямз Р. Операционная система OS/2: Пер. с англ. - М.: Мир, 1991. - 351 с.
10. Орлик С.В. Секреты DELPHI на примерах. М.: Восточная книжная компания, 1996. - 352 с.