

**Тема: Основні правила проектування діалогу з користувачем, області команд.**

**Мета:**

**Завдання до комп'ютерного практикуму**

Використання функціональних кнопок та правил проектування діалогу:

1. Спроекувати форми інтерфейсу з використанням кнопок навігації згідно правил проектування області функціональних клавiш.
2. Спроекувати форми інтерфейсу з використанням кнопок продовження діалогу з користувачем: „Дія відмовлення”, „Дія введення”, „Дія – вихід”, „Дія – довідка”, „Дія – підказка”, „Дія – регенерація” згідно правил проектування області функціональних клавiш.
3. Пояснити застосування правил на прикладі спроектованої форми або форм.

Використання функціональних кнопок та правил проектування області команд:

4. Спроекувати форми інтерфейсу з використанням компонентів введення інформації щодо подання команд користувачем з наступною реакцією додатку:
  - 1.1. запуск зумовленої процедури на виконання з завданням параметрів в області команд;
  - 1.2. виклик простого побудувала виразів для формування команди і наступного її виконання.
5. Спроекувати форми інтерфейсу для реалізації дії „команда” за методом виклику додаткової форми в окремому вікні.

**Питання:**

1. Призначення, розміщення, зміст області команд.
2. Правила запиту і виконання дії “Команда”.
3. Правила для дій “Витяг команди” та “Підказка”.
4. Співвідношення між областю команд і меню дій
5. Призначення, розміщення, геометричне зображення, категорії кнопок (функціональних клавiш).
6. Кнопки навігації: застосування на формах і панелях, склад, реалізація дій.
7. Кнопки продовження діалогу: дія “введення”, дія “відмовлення”, дія “вихід”.
8. Кнопки продовження діалогу: дія “підказка”, дія “довідка”, дія “регенерація”.

## Короткі теоретичні відомості

### Основні поняття, що використовуються при проектуванні діалогу.

**Діалог** – це послідовність запитів і відповідей, якими обмінюються користувач і машина. Увесь діалог заснований на уніфікованих діях (ввод, вибір, відмова, вихід).

**Ввод** потрібен для того, щоб машина могла виконувати дії згідно алгоритму. А користувач повинен прочитати інформацію на екрані, яка виведена машиною, щоб він зміг провести аналіз, прийняти рішення і продовжити роботу з додатком (запис до БД, виведення звіту на екран).

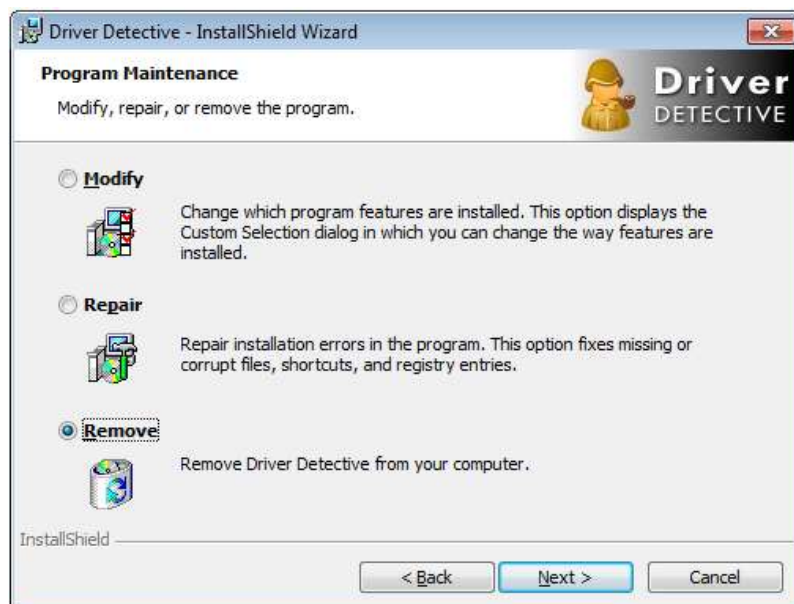
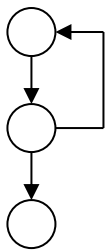
**Вибір** аналог вводу, але різниця в тому, що машина видає пропозиції користувачу і серед списку пропозицій він повинен обрати одну, яка на його думку є головною чи необхідною на певному кроці виконання задачі.

**Відмова** – це дія, яка пов'язана з переходом на попередній етап, пункт, точку роботи додатку. Це може бути в межах форми або повернення на попередню форму (кожні дві форми поки ми не перейшли на третю, містять свою інформацію в буфері).

**Вихід** – це аналог відмови, але він припускає повернення в найвищу точку алгоритму додатку або операційного середовища.

При реалізації діалогу інформація може відображатися на екрані деякий проміжок часу для оцінки її користувачем. Користувач може змінити цю інформацію і потім зберегти її, а іноді ця інформація може змінюватися, але не бути збережена. Додаток повинен робити запит для підтвердження збереження чи незбереження інформації.

Кнопки продовження діалогу з користувачем;



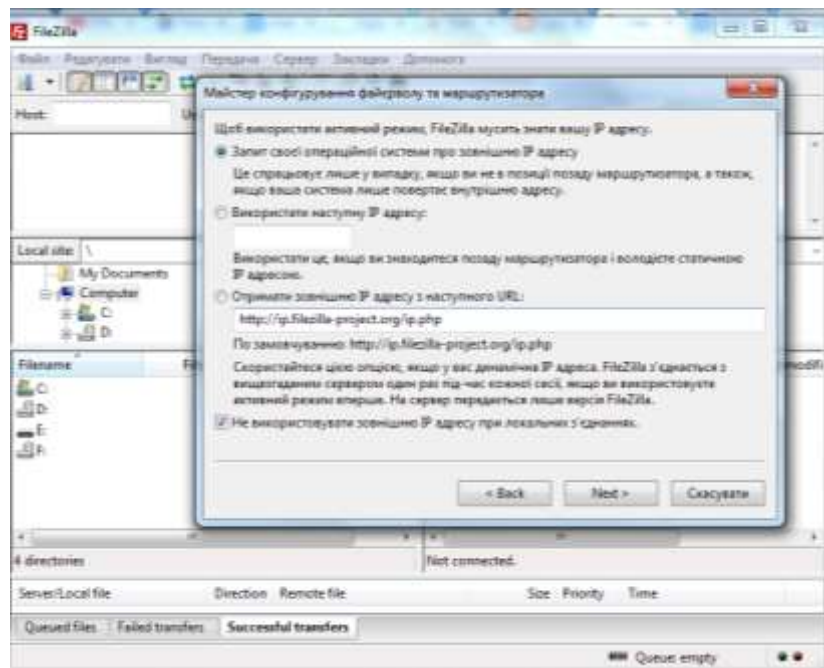
Кнопки

продовженн

Мал 137 Використання кнопок продовження діалогу з користувачем

**Область команд: розміщення, зміст, використання, правила запиту дії “Команда”.**

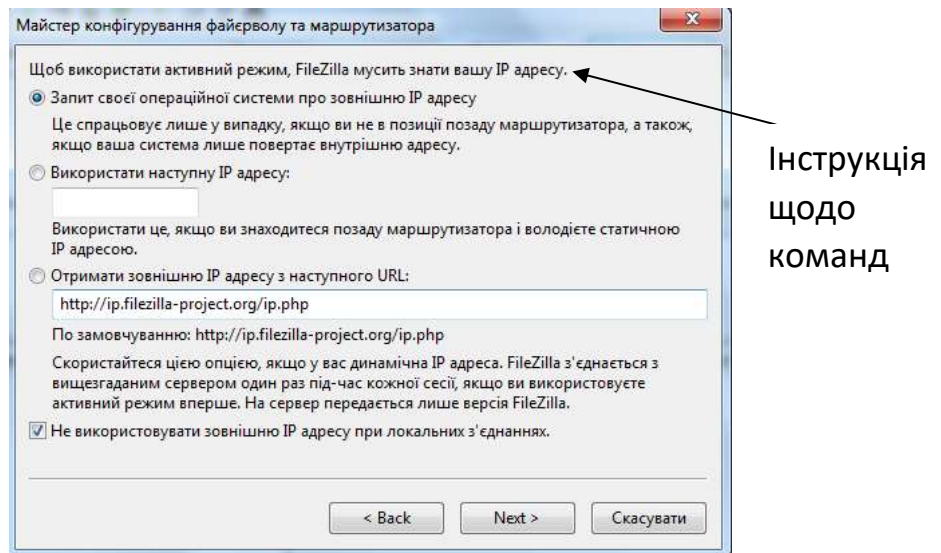
Розташування області команд може бути в окремій частині форми, панелі або екрану. При розміщенні вікон для запису команд може трапитись таке, що необхідний виклик допомоги або підказки для написання цієї команди. В цьому випадку незалежно від кількості довідкової інформації, яка буде виведена на екран області команди, область повинна бути доступною для запису команди користувачем, тобто переключення вікон панелі, виклик додаткової інформації не повинні затуляти цю область. Область команд може містити заголовки, якщо сама назва команди не досить інформативна. Ряд послідовних команд можна об'єднати або згрупувати в якусь групу, в цьому випадку доцільно використовувати заголовки. Все тіло області команд повинно вміщувати компоненти введення інформації. На різноманітні команди користувача повинна бути реакція додатку. Тобто бажано здійснити відображення результатів цих дій в окремій області форми, або в окремих вікнах.



Мал 125 Задання різноманітних команд користувача.

Ця інформація повинна доводитись до користувача в формах, які не потребують додаткового розтлумачення. Користувач може запросити чи викликати дію або команду в діалозі з додатком. Існують певні правила запиту і виконання дії “Команда”:

1. При запиті дії “Команда” самі засоби подання команди (компоненти) можуть з’являтися у вторинних вікнах або додаткових вікнах. Текстовий курсор встановлюється на першу позицію компоненти введення команди, що служить звертанням уваги користувача на можливість введення команди.
2. Якщо для подачі команд з’являється вторинне або додаткове вікно, то увага користувача може попередньо бути звернена на інструкцію щодо подання команд. Ця інструкція записується попередньо від області команд (зверху).

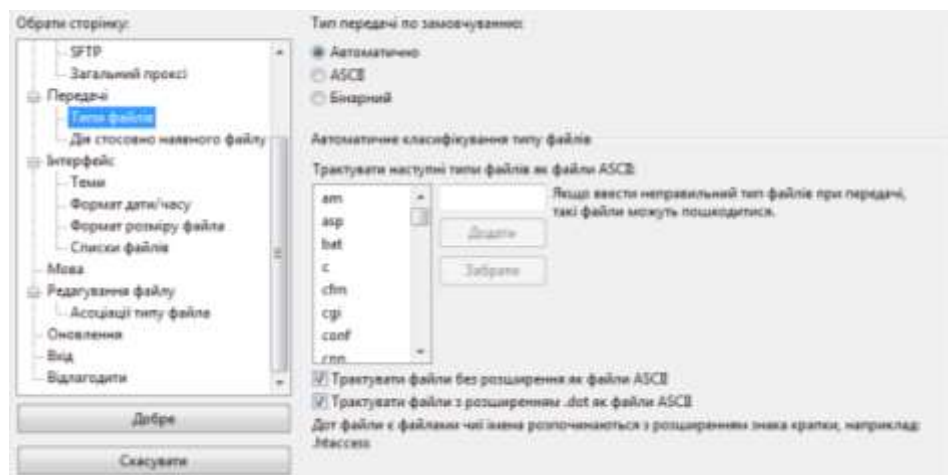


Мал 127 Відображення (зверху) інструкції щодо подання команд

Професійний рівень додатку буде тим вище, чим більше введено розробником різноманітних засобів настроювання і адаптації, тобто, якщо у випадку налагодження і адаптації команд буде зроблено внутрішній інспектор коду і систему підказок або довідкову систему щодо правильності написання команд, то це приведе до зростання ефективності дій додатку.



Мал 128 Введення різноманітних засобів настроювання додатку.



Мал 129 Введення настройки функціональності додатку для зручності роботи.

При натисненні ↵ (enter) дії додатку продовжуються згідно алгоритму, тобто зникають всі інформаційні вікна і курсор встановлюється на наступний компонент (на компоненти введення, кнопки або на подання інструкцій щодо використання клавіш).

3. В деяких додатках застосовується метод виклику додаткової форми в окремому вікні, яка спеціально призначена для подання команд при цьому:

- 1) В кожній формі, яка потребує виклику цього вікна, повинна бути кнопка з викликом цієї команди.
- 2) Зміст і прядок використання цього вікна повинні бути однаковими (хоча зміст інструкцій, синтаксис і параметри можуть бути різними).

В додатку може бути запрограмований випадок, коли користувач може відходити від жорсткої схеми алгоритму взаємодії з додатком. При цьому для користувача повинні бути надані засоби виклику спеціального вікна подачі команди різноманітних дій. Наприклад, він може облишити на деякий час вирішення задач додатку, перейти в інше середовище або викликати на виконання якусь програмну оболонку, або звернутися до довідкової інформації. В цьому випадку робота в поточному додатку буде призупинена. При цьому може бути згортання поточної форми і всього додатку. В статусному рядку середовища виклику повинна розташовуватись позначка (кнопка) повернення до цього додатку. Можна запрограмувати комбінацію клавіш (Ctrl+.....) для згортання цього додатку і необхідності роботи в середовищі, з якого був викликаний цей додаток.

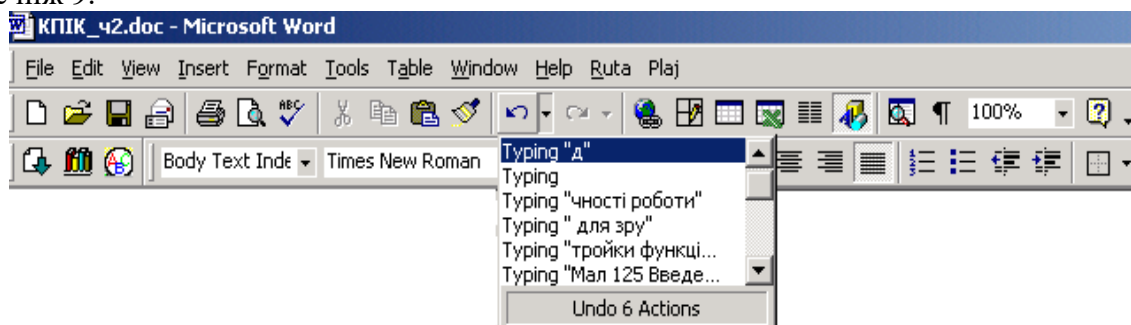
## **Правила для дій “Витяг команди”, правила для дії “Підказка”.**

Користувач може не тільки записувати нову команду, але й використовувати одну з наданих в минулому команд. Для цього система повинна містити перелік команд, які вже були подані і виконані. Остання важлива з точки зору перевірки синтаксису команд, і в результаті виконання цих команд траплялись якісь події, які користувач бачив і проаналізував, тому він пам’ятає дії команд і може витягнути із списку будь-яку із них. Правила дії “Витяг команди” наступні:

1. При запиті дії “витяг команди” на екран в області команд може бути виведена остання команда або остання команда зі списком команд, що відкривається, або зразу листи деякої кількості команд. При пректуванні треба знати, що виведення однієї команди виконується в разі частого редагування параметрів і повторення однієї і тієї ж самої команди. Витяг зі списком - в разі обмеженої кількості команд, що подаються, а повний список – в разі майже випадкового використання будь-якої з попередньо поданих команд.

2. Витягнуту команду можна змінити, додати чи замінити параметри, або просто запросити її виконання. Останнє виконується, як правило, для команд, котрі здебільшого не потребують задання параметрів . якщо проводиться дослідження і необхідно часто змінювати тип параметрів однієї команди, то курсор зразу повинен встановлюватися на перший параметр, що може змінюватись.

3. При багаторазовому запиті дії “Витяг” на екран, можуть зберігатись всі попередньо подані команди. Для цього в системному додатку повинен вестись спеціальний протокол подання команд на виконання. А для того, щоб була виконана потрібна команда, в полі розміщується список команд із скролінгом. Кількість команд не повинна сягати більше ніж 9.



Мал 130 Приклад ведення протоколу подання команд на виконання

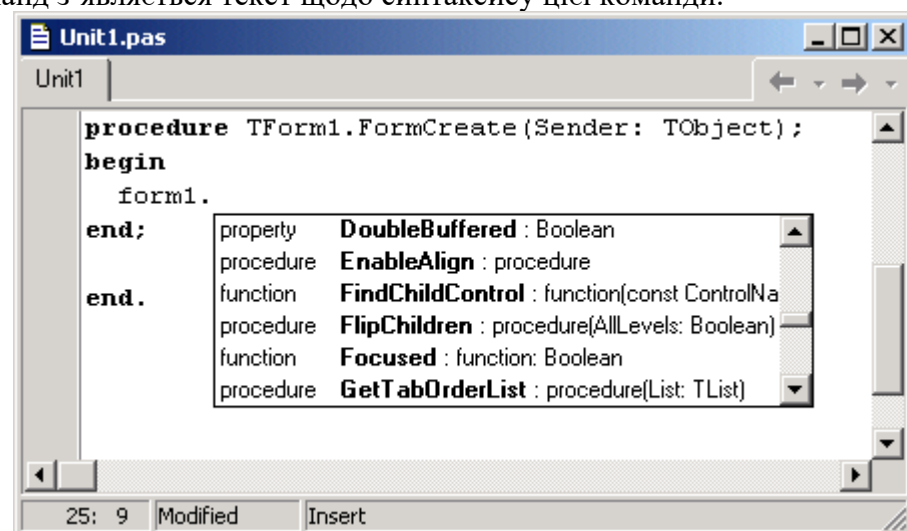
Підказка потрібна користувачу для того, щоб зрозуміти якусь абривіатуру або скорчення, дати тлумачення піктограмі або слову, яке не часто зустрічається в тексті, або надати інформацію щодо поточного вигляду, або дії, або інформаційних повідомлень, також щодо області де знаходиться курсор.

Відображення  
підказки на  
екрані



Мал 131 Реалізація режиму підказок.

Це означає, що можна спроектувати підказку в будь-якій області форми, панелі або екрану. Підказка з'являється тільки тоді, коли на цю область наведений курсор і покажчик (стрілка) затримується на цій області на деякий час. Підказка може використовуватись і при формуванні дії “Витяг команди”, в цьому випадку при розташуванні курсору в області параметрів команд з'являється текст щодо синтаксису цієї команди.

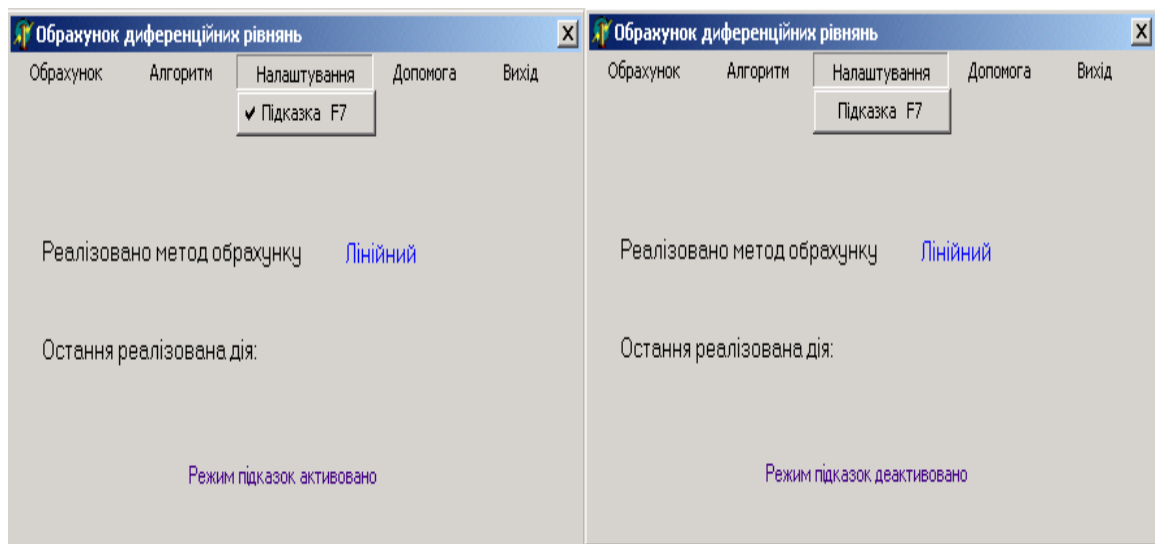


Мал 132 Підсвічування синтаксису команди.

Правила для формування дії “Підказка” наступні:

1. Текст підказки треба розмішувати в окремому віконці з різними шрифтами, кольором фону, щодо основного тексту форм, панелі або екрану.
2. Текст підказки повинен бути якомога коротшим і не мати незрозумілих слів.
3. Користувач повинен мати можливість щодо відключення як всіх, так і деякої частини підказок. Це відноситься до елементів налаштування або адаптації додатку.





Мал 133 Реалізація можливості в додатку відключення підказок

### Співвідношення між областю команд і меню дій.

Меню дій складається з опцій, можна сказати, загальних команд. Ці команди мають узагальнене призначення і частіше за все приводить до запуску на виконання якоїсь великої задачі чи підзадачі.

Область команд розташовується у формі і має обмежене призначення: виконання якоїсь дії після подання команд у конкретній задачі.

Меню дій ніколи не має опцій з параметрами, а область команд має можливість подання команд з параметрами. Для проектування області команд при існуючих меню дій треба виконувати наступні правила:

1. Меню дій і область команд не повинні взаємно виключати один одного.
2. Меню дій не повинно містити всі команди, які можуть бути використані в області команд. Якщо вона використовує систему команд, то треба, щоб користувач подавав їх в області команд, а не в області меню дій, тому що меню дій має функціональне навантаження (тобто те, що відноситься до предметної області). Якщо додаток потребує виконання деяких відомих в інших оболонках системних операцій (виклик файлів, робота з ними, редагування...), то їх треба розміщувати, як опції меню дій.
3. Якщо деякі команди можна викликати з меню дій і з області команд, то вони повинні однаково називатися.
4. Доцільно усі узагальнені функції формувати, як опції меню дій, а параметри деталізувати в різних спливаючих вікнах.
5. Ступінь деталізації дії, що виконується за допомогою задання команд, повинна відповідати правилу сімки – тобто для виконання якоїсь дії користувач не повинен використовувати більше ніж 7 операторів або команд.