

Створення контекстно-залежної системи допомоги для прикладних програм

Поява великої кількості різноманітних програмних продуктів та їх значна складність потребує наявності в них розвинутої системи допомоги. Така система повинна бути контекстно-залежною (тобто викликатись повинен саме той розділ допомоги, який потрібен в даний момент для роботи з певним елементом програми), дозволяти пошук необхідного матеріалу як за назвами розділів, так і за ключовими словами, містити посилання на інші розділи системи допомоги або пояснення до окремих термінів.

Серед різноманітних систем допомог для програмних засобів, що використовуються в операційних системах DOS та Windows, можна виділити такі три основні типи:

- система допомоги, вбудована в конкретний програмний продукт, з додержанням внутрішньої структури цього продукту. Файл допомоги такої програми не може бути прочитаним без використання самої програми. Прикладом може служити ППЗ Gran1 для DOS;
- файл або файли системи допомоги, що викликається окремою програмою – так званим “в’ювером” (від англійського View – бачити), призначеною для роботи з файлами певної структури і незалежною від самого програмного продукту. Такий *help*-файл може бути прив’язаний до будь-якої програми. Прикладами є стандартні файли допомоги ОС Windows і

Microsoft Office 2000. Для їх перегляду створені спеціальні програми Winhelp.exe та Msohelp.exe відповідно;

- HTML-файли, які зазвичай переглядаються за допомогою будь-якого із броузерів, що проінстальовані на комп'ютері користувача.

Звичайно, що перший тип файлів є найбільш незручним. З інших двох кожний має свої переваги і недоліки. Так, найпростішим для створення є гіпертекстовий HTML-файл, але в ньому неможливий як такий пошук за назвами розділів (за змістом) чи за ключовими словами. В той же час *help*-файли другого типу містять додаткові можливості пошуку потрібної інформації, але їх створення пов'язане з певними труднощами. Оскільки створення гіпертекстового файлу в HTML-форматі хоча і містить певні особливості, але не викликає суттєвих проблем, хотілося б докладніше зупинитись на створенні системи допомоги в стандарті Windows. На жаль, як сама операційна система Windows, так і всі програмні засоби, які призначені для створення систем допомоги, існують лише в англійській та російськомовній версіях, тому посилання на різні пункти та послуги даються мовою оригіналу.

Основне, на що потрібно звернути увагу при створенні *help*-файлу в стандарті Windows – формат представлення даних. Система допомоги складається з двох основних файлів: відкомпільованого файлу *довідки* *.HLP та тестового файлу *змісту довідки* *.CNT. Файл допомоги автоматично викликається з програмного продукту при натисненні на клавішу F1 або обранні пункту меню “Допомога” (“Help”). При цьому виконується програма Winhelp.exe або Winhlp32.exe, в головне вікно якої і завантажується відповідна сторінка файлу допомоги (рис.1) в залежності від того, де в даний час знаходиться користувач, тобто пояснення до якого розділу він хоче одержати. Службовий файл *.GID система Windows створює автоматично при першому зверненні до довідки.

Організацію системи допомоги умовно можна розділити на такі етапи:

- 1) побудова текстового файлу *довідки* (написання тексту кожного розділу, визначення ідентифікаторів кожного розділу, організація гіперпереходів);
- 2) побудова текстового файлу *змісту довідки*. Хоча зміст і є необов'язковим, але він значно полегшує пошук необхідного розділу;
- 3) створення за допомогою спеціального компілятора файлу *проекту*;
- 4) компіляція довідникової системи;
- 5) підключення одержаного файлу до програмного засобу та організація виклику відповідних тем.

Файл довідки створюється з двох фізичних файлів:

- файлу форматowanego тексту, який містить текст довідкової системи. Найчастіше це файл в форматі RTF;
- текстового файлу проекту *.HPJ, в якому вказані опції компілятора.

Перше, з чого потрібно розпочати створення системи допомоги, – написати текст довідкової системи. Його можна створити в будь-якому редакторі, якій підтримує формат RTF та деякі з засобів форматування тексту (виноски, підкреслення, схований текст тощо). Найпростіше за все – скористатись текстовим процесором Microsoft Word. Однак треба мати на увазі, що, на жаль, різні версії текстового процесору Microsoft Word створюють різні за внутрішньою структурою файли формату RTF. Результатом цього є те, що деякі компілятори не “розуміють” тексти, створені в останніх версіях Word’у. Так, компілятор Microsoft Help Workshop, що входить в стандартний комплект поставки Delphi 3.0, компілює файли, записані в RTF-форматі з використанням Microsoft Word

6.0, але вже не в змозі цього робити, якщо файл створений навіть в версії Microsoft Word 7.0.

При створенні тексту довідки потрібно знати наступні поняття:

- *розділ* – фрагмент довідникової системи, що відповідає окремому елементу або певній темі програмного засобу. У вікні програми Winhelp.exe завжди відображається лише один розділ (рис.1);

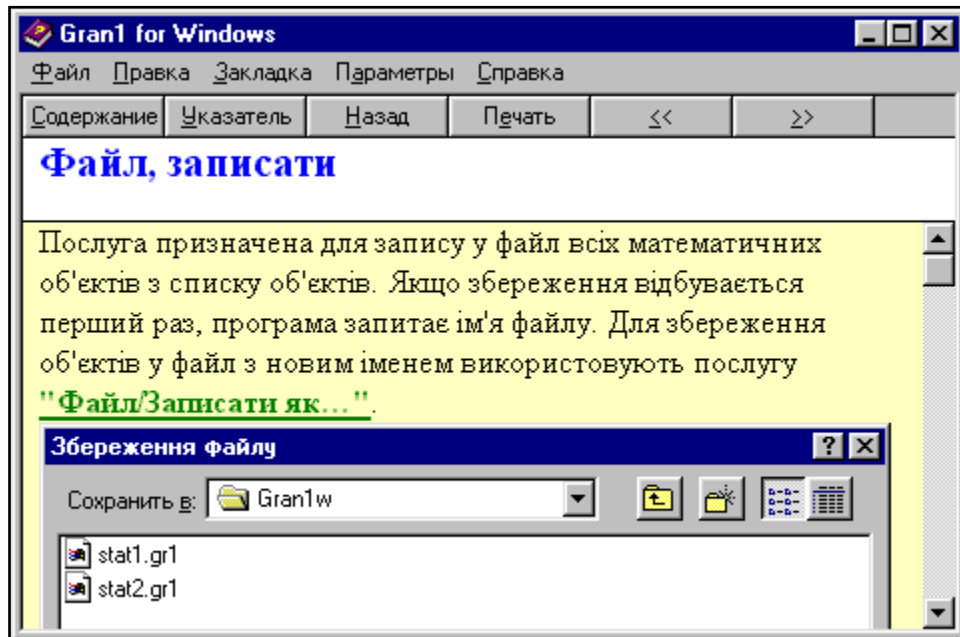


Рис. 1

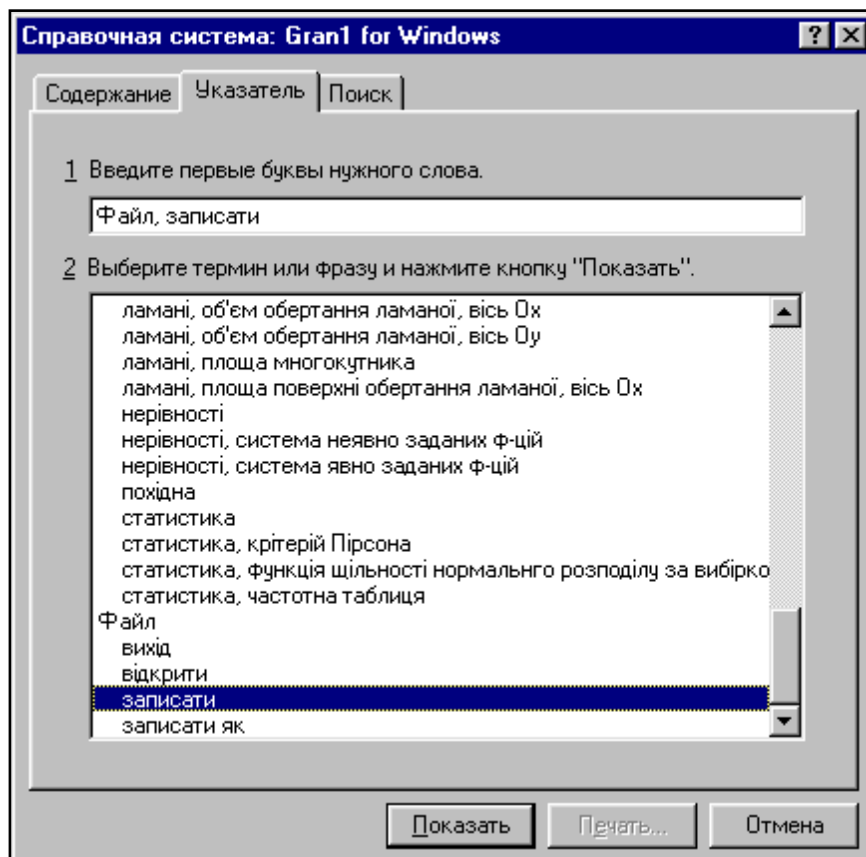


Рис. 2

- **ідентифікатор розділу** – текстовий рядок, який однозначно визначає (ідентифікує) розділ. Він не повинен перевищувати 255 символів;
- **ключове слово** (слова) – слово, за якими буде відбуватись пошук в розділі довідки “Указатель” (рис.2);
- **тема розділу** або **заголовок** – назва, під якою розділ з’являється у вікні довідки в процесі пошуку потрібного розділу (рис.3);

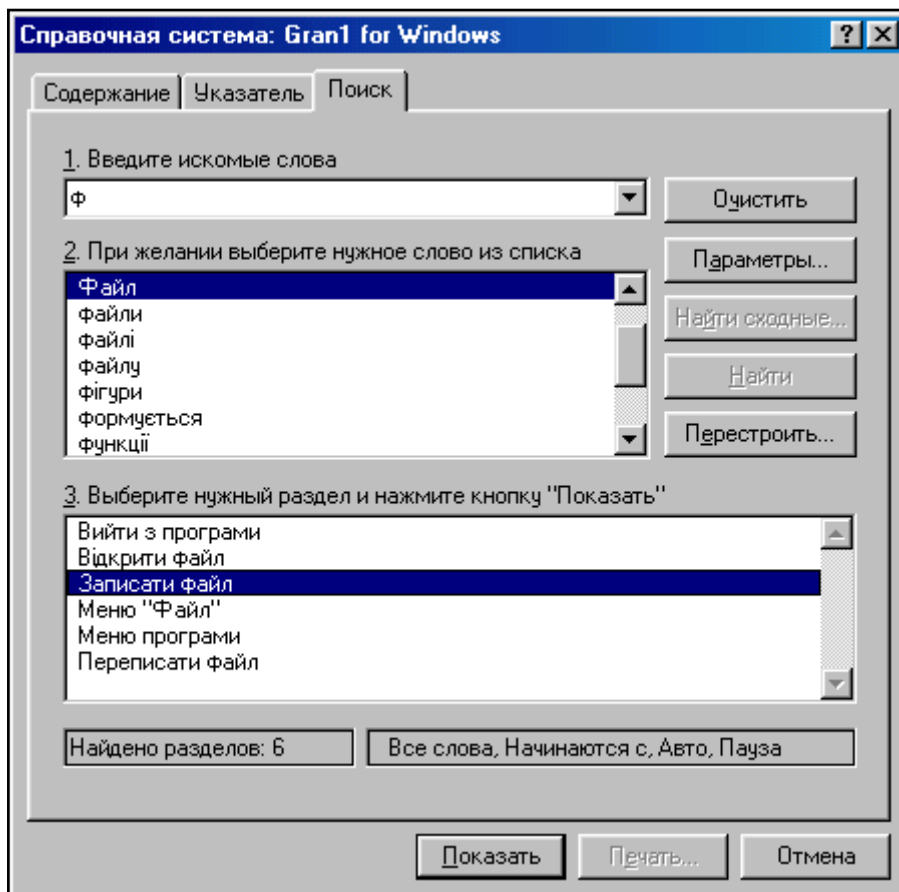


Рис. 3

- **номер розділу** – номер, за яким розділ буде пов’язаний з програмним продуктом. Також вказує в якому порядку повинні виводитись теми при пострінковому перегляді довідки з використанням клавіш переходу. Порядок такого перегляду може не співпадати з порядком сторінок в RTF-файлі, який містить текст довідникової системи.

Створювати текст треба таким чином, щоб кожна сторінка тексту являла собою окреме вікно (розділ) системи допомоги, що відповідає

певній частині або елементу програмного засобу. Така сторінка повинна містити заголовок, відповідний текст, посилання на інші розділи довідки, якщо вони необхідні, та виноски. Кожна, без винятку, сторінка (в тому числі остання) повинна містити в кінці символ розриву сторінки. Весь текст можна урізноманітнити шрифтами, кольорами та малюнками або посиланнями на малюнки (рис.4).

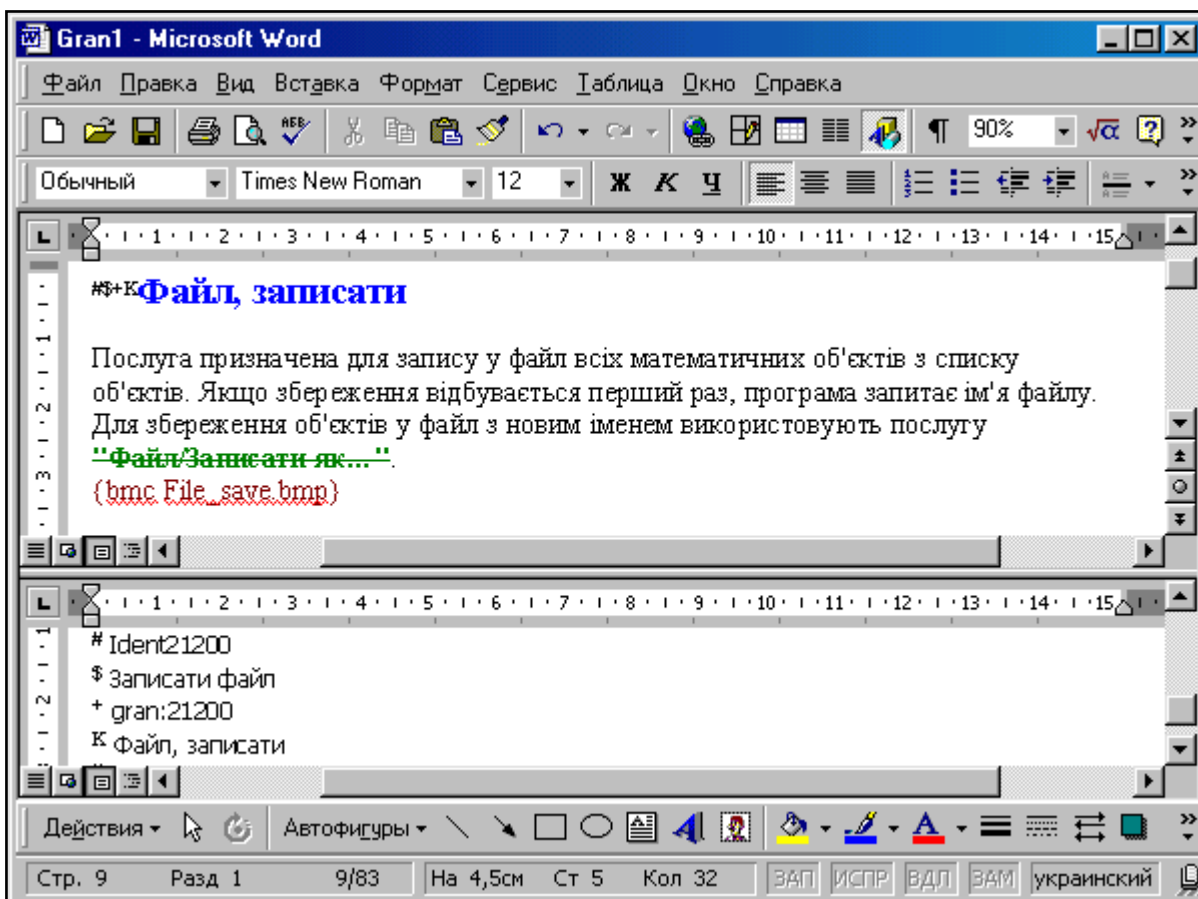


Рис. 4

Виноски являють собою ідентифікатори, що надалі використовуються при компіляції системи. Зазвичай виноски розташовують перед першим словом заголовку сторінки, а всі вони повинні бути звичайними (**Вставка/Сноска.../Обычная** в процесорі Word), тобто власне текст виносок розміщується наприкінці тієї ж сторінки. Найчастіше використовують такі символи для виносок (дивись рис.4):

- # ідентифікатор розділу (“*Ident21200*” на рисунку);
- \$ тема розділу (“*File, save*”);

- К ключове слово або слова (“*Файл, записати*”);
- + номер розділу (“*gran:21200*”, однак фактично, номером розділу є тільки число 21200).

Обов’язково повинна бути наявною лише виноска ідентифікатору розділу. Всі інші виноски необов’язкові, але з ними система буде мати більше можливостей.

Організація гіперпереходів відбувається наступним чином:

- слово або словосполучення, до якого під’єднується гіперпосилання на інший розділ довідникової системи, виділяється закресленим шрифтом (рис.4). При виклику у вікні відбувається перехід до нового розділу;
- слово або словосполучення, до якого необхідно подати пояснення за допомогою спливаючої підказки виділяється підкресленим шрифтом. При виклику текст у вікні допомоги не змінюється, а додатково з’являється невелике віконце, що містить пояснюючий текст до терміну. Цей текст зникає при натисненні на будь-яку клавішу клавіатури або кнопку миші;
- одразу після слова, без пропусків, пишеться ідентифікатор розділу, на який відбувається перехід. Сам ідентифікатор повинен бути написаний схованим шрифтом. Так на рисунку 4 після перекресленої фрази “*Файл/Записати як...*” повинно бути написане “*Ident21300*”. Але, оскільки ідентифікатор пишеться схованим шрифтом, то на малюнку він не відображений.

Зміст довідки містить в собі назви розділів, які згруповані за деяким принципом. Він може створюватись за допомогою будь-якого текстового редактору, наприклад “Блокнот”. Перед кожним рядком ставиться число, яке визначає рівень підпорядкування даного рядка. Якщо рядок визначає перехід на розділ довідникової системи, то після назви ставиться знак рівності і ідентифікатор відповідного розділу. В іншому випадку рядок

визначає назву групи. Наведемо фрагмент такого файлу змісту, відкомпільований результат якого представлено на рис.5.

```
:BASE GRAN1.hlp
:TITLE Gran1 for Windows
1 Вікна програми
2 Вікно "Список об'єктів"=Ident11000
2 Вікно "Графік"=Ident12000
2 Вікно "Відповіді"=Ident13000
1 Пункти меню
2 Файл
3 Відкрити...=Ident21100
3 Записати=Ident21200
3 Записати, як...=Ident21300
3 Вихід=Ident21400
```

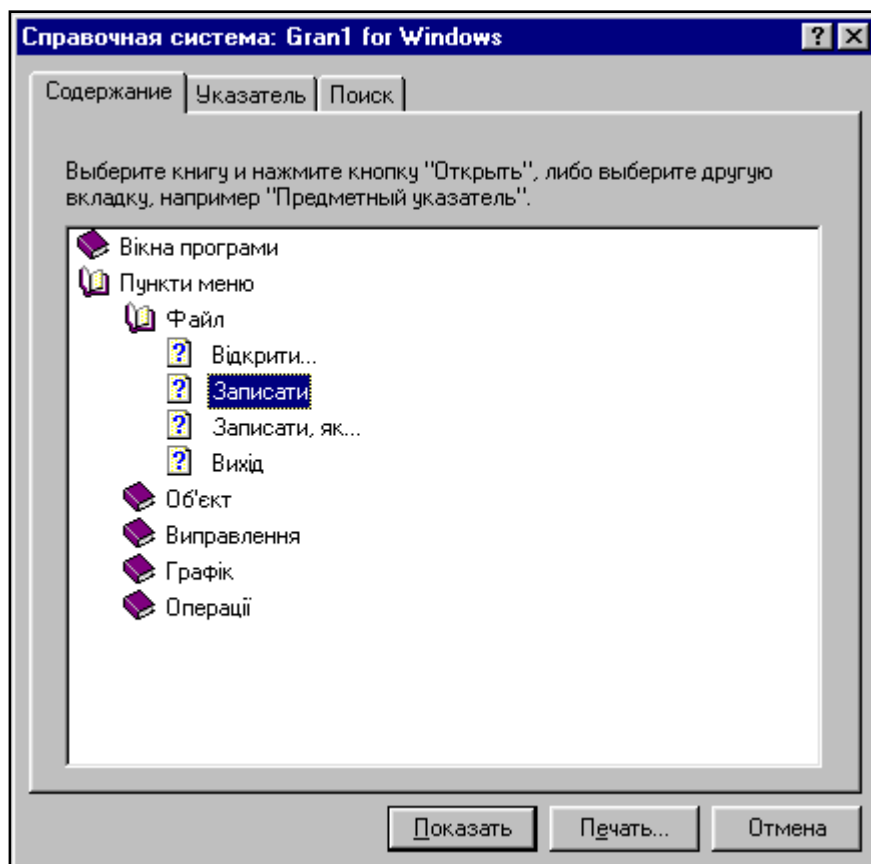


Рис. 5

Для створення файлу проекту та компіляції всіх файлів в єдиний файл системи допомоги необхідно скористатись спеціальною програмою-

компілятором, наприклад Microsoft Help Workshop. Потрібно вказати назву файлу-результату, назви файлів тексту та змісту довідки в розділі “**Option\Files**”, а також, за бажанням, додаткові параметри, наприклад, розмір вікна та його розташування при виклику, кольори, наявність додаткових кнопок тощо. Створений файл проекту компілюється в файл допомоги *.HLP та може під’єднуватись до програмного продукту.

Як саме під’єднати файл допомоги, залежить від середовища програмування. Так, в Borland Delphi потрібно в розділі “**Project\Options...\Application\Help file**” вказати ім’я файлу довідки, властивості “**Help file**” та “**Help context**” об’єкту, для якого необхідно викликати довідку, змінити на ім’я файлу довідки та номер розділу відповідно.

Як бачимо, створення такого контекстно-залежного файлу допомоги не є простою справою. Тим більше, що робити це доводиться за допомогою різних програмних продуктів одночасно. Саме тому в останній час з’явилися інтегровані середовища для створення *help*-файлів. Такі середовища мають власний текстовий редактор із можливостями форматування тексту та організації гіперпереходів, редактор змісту довідки, компілятор та засоби експорту та імпорту даних і ще багато додаткових можливостей. Серед них слід виділити наступні:

- **HelpScribble** – найпростіший серед всіх програмних продуктів даного типу і може бути рекомендований для користувачів початкового рівня, особливо для тих, що програмують в Delphi (рис.6). Всі елементи майбутньої системи допомоги об’єднуються в один файл проекту, який потім легко компілюється в HLP-файл в форматі WinHelp 3.0 або WinHelp 4.0 (тобто в форматі Windows 3.1 або Windows 95 відповідно). Програма має засоби експорту-імпорту даних в RTF- та CNT-файли для опрацювання в інших системах.
- **Help Studio 2000** – дещо складніша програма, що має практично такі самі можливості під час написання файлу проекту. Основні відміни полягають в підтримці декількох мов, додатково компіляції в

форматах Windows CE Help та HTML Help. Цікавою особливістю є можливість імпортувати дані з існуючого файлу довідкової системи (декомпіляція HLP-файлу).

- **RoboHELP Explorer** – призначений для професійних розробників програмного забезпечення. При інсталяції інтегрується в пакет Microsoft Office (95, 97, 2000), має багато можливостей, включаючи елементи мультимедіа, графіку, WEB та інше, орієнтований на створення систем допомоги для більшості основних засобів програмування (Borland Delphi, Borland C++, Turbo Pascal, Visual Basic, Visual FoxPro тощо), підтримує всі відомі на даний час формати HLP-файлів для ОС Windows.

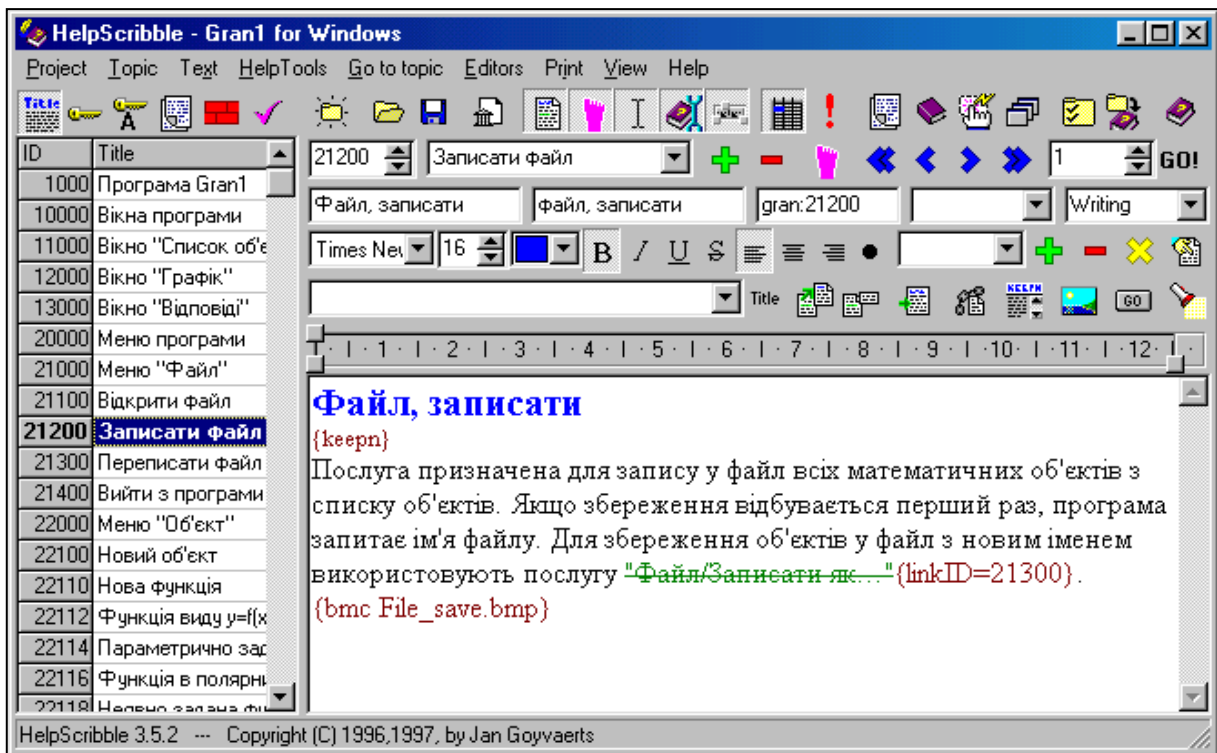


Рис. 6

Таким чином, можна сказати, що написання системи допомоги, інтегрованої в програмний продукт, є невід'ємною частиною створення самої програми. Наявність її не тільки допомагає користувачу швидше та якісніше оволодіти навичками роботи з прикладними програмами, а і висвітлює рівень професіоналізму та інформаційної культури розробників програмного забезпечення.

Література

1. Гофман В.Э., Хомоненко А.Д. Delphi 5. – СПб:БХВ – Санкт-Петербург, 2000. – 800с.:ил.
2. Сван Т. Основы программирования в Delphi для Windows 95. – К.: “Диалектика”, 1996. – 480с.:ил.