

Лекція 21.

Модулі

*Мельник О.О.
доцент кафедри
програмного забезпечення систем*



План:

- 1. Модульне програмування.**
- 2. Структура модуля.**
- 3. Компіляція і використання модулів.**
- 4. Стандартні (вбудовані) модулі.**
- 5. Приклад побудови модуля користувача.**



Рекомендована література:

1. Turbo Pascal 7.0: Пер. с нем. / Винфрид и Фолькер Кассера. – К.: Издательство «ДиаСофт», 2003.
2. Грызлов В.И., Грызлова Т.П. Турбо Паскаль 7.0. – 4-е изд., испр. – М.: ДМК Пресс, 2005.
3. Паскаль для персональных компьютеров: Справ. Пособие / Ю.С. Бородич, А.Н. Вальвачев, А.И. Кузьмич. – Мн.: Выш. Шк.: БФ ГИТМП «Ника», 1991.
4. Паскаль. Программирование на языке высокого уровня: Учебник для вузов / Т.А. Павловская. – СПб: Питер, 2003.
5. Программирование в среде Turbo Pascal 7.0 / Марченко А.И., Марченко Л.А.: Под ред.: Тарасенко В.П. – 8-е изд. – К.: ВЕК+, СПб: КОРОНА принт, 2004.
6. Зеленьяк О.П. Практикум программирования на Turbo Pascal. Задачи, алгоритмы и решения. – СПб.: ООО «ДиаСофтЮП», 2002.



Модульне програмування

На мові Паскаль складна задача може бути розбита на більш прості за допомогою *підпрограм* (процедур та функцій). Після цього програму можна розглядати у більш укрупненому вигляді – на рівні взаємодії підпрограм.

Використання підпрограм є першим кроком до підвищення ступеня абстракції програми і призводить до спрощення її структури.



Модульне програмування

Розбиття програми на підпрограми дозволяє запобігти надмірності коду, оскільки підпрограму записують один раз, а викликати її можна багаторазово з різних точок програми.

Наступним кроком у підвищенні рівня абстракції програми є групування підпрограм і пов'язаних з ними даних в окремі файли (**модулі**), які компілюються окремо.



Модульне програмування

Модуль – це бібліотека ресурсів, котра підключається до програми.

Модуль може містити опис типів, констант, змінних і підпрограм. У модуль, як правило, об'єднують зв'язані між собою ресурси.

Деталі реалізації модуля приховані від користувача. Модулі зберігаються у готовому, скомпільованому вигляді. Це підвищує швидкість загальної компіляції програми.



Модульне програмування

Після компіляції модулі зберігаються на диску в файлах з розширенням **.TPU** (*Turbo Pascal Unit*).

Модулі на мові Паскаль поділяються на:

- стандартні;
- модулі користувача.



Структура модуля

Unit <ім'я модуля>;

Interface {інтерфейсна секція}

<заголовок підпрограми 1 із указанням параметрів>

<заголовок підпрограми 2 із указанням параметрів>

...

<заголовок підпрограми N із указанням параметрів>

Implementation {секція реалізації}

Uses <імя модуля1>, ... ;

<заголовок без параметрів і тіло підпрограми 1>

...

<заголовок без параметрів і тіло підпрограми N>



Структура модуля

begin {секція ініціалізації}

[<оператор 1>;

...

<оператор N>]

end.



Структура модуля

Розділ **Unit** містить ім'я модуля. Воно повинно співпадати з іменем файлу на диску (з розширенням pas), в якому міститься вихідний текст модуля.

Наприклад:

у файлі **Start.pas** міститься текст модуля
Start:

Unit **Start**;



Структура модуля

У *інтерфейсній секції* описуються *глобальні* константи, типи, змінні, процедури і функції. Основна програма має доступ до цих даних без додаткового їх опису за умови використання модуля.

У *секції реалізації* визначаються модулі всіх глобальних процедур та функцій. Описуються *локальні* константи, змінні, процедури і функції, які недоступні основній програмі.



Структура модуля

Секція реалізації повинна бути останньою секцією модуля.

Вона може складатися:

або

тільки із зарезервованих слів **begin** і **end** (у цьому випадку модуль не містить коду ініціалізації),

або

з **операторної частини**, котра повинна виконуватися для ініціалізації модуля.



Компіляція і використання модулів

У середовищі Turbo Pascal визначені три режими компіляції: *Compile, Make, Build*.

При компіляції модуля або основної програми у режимі **Compile** всі підключені модулі (у розділі Uses) повинні бути попередньо відкомпільовані у однойменні файли з розширенням .tpr .



Компіляція і використання модулів

У режимі **Make** компілятор перевіряє наявність три-файлів для кожного підключеного модуля. Якщо один з файлів не знайдено, система намагається відшукати однойменний файл з розширенням `.pas` (файл з вихідним текстом модуля) і, якщо такий файл знайдено, приступає до його компіляції.

Крім цього система слідкує за можливими змінами вихідного коду будь-якого модуля, котрий використовується, і при наявності змін автоматично його перекомпільовує.



Компіляція і використання модулів

У режимі **Build** існуючі три-файли ігноруються, і система намагається відшукати відповідний pas-файл для кожного оголошеного в розділі Uses модуля та провести його компіляцію.



Компіляція і використання модулів

Порядок компіляції модулів

1. У середовищі мови Паскаль відкрити pas-файл з кодом модуля (*вважаємо, що помилки у тексті відсутні*).
2. У головному меню встановити режим **Compile/Destination/Disk**.
3. Виконати компіляцію, використовуючи режим **Compile**.
4. При коректному виконанні пп. 1-3 на диску автоматично формується відповідний три-файл.



Компіляція і використання модулів

Готові модулі (сформовані три-файли) можна використовувати у програмах. Для цього їх імена необхідно записати у розділі опису модулів, що використовуються, основної програми (**Uses**).

Якщо файли модулів розміщені не у робочому каталозі, то шлях до них необхідно вказати у головному меню середовища мови Паскаль:
[Options/Directories/Unit Directories](#).



Стандартні (вбудовані) модулі

У мові Turbo Pascal є дев'ять стандартних модулів:

Назва модуля	Короткий опис
System	Містить засоби для файлового введення/виведення, обробки рядків, операцій із плаваючою комою, динамічного розподілу пам'яті, а також процедури і функції, які не ввійшли до інших модулів.
Crt	Призначений для ефективної роботи з екраном у текстовому режимі, клавіатурою та вбудованим динаміком



Стандартні (вбудовані) модулі

Стандартні модулі Turbo Pascal:

Назва модуля	Короткий опис
Printer	Забезпечує вивід інформації на принтер
Strings	Призначений для роботи з рядками, які закінчуються нульовим символом
Overlay	Містить засоби для управління оверлеями
Dos	Містить підпрограми, які реалізують можливості операційної системи MS-DOS



Стандартні (вбудовані) модулі

Стандартні модулі Turbo Pascal:

Назва модуля	Короткий опис
Graph	Забезпечує роботу з екраном у графічному режимі, дозволяє формувати графічні зображення і виводити на екран текстові надписи.
Turbo3 Graph3	Містить засоби для сумісності з версією 3.0 Турбо Паскаля



Стандартні (вбудовані) модулі

Стандартні модулі **Graph**, **Turbo3**, **Graph3** містяться **в однойменних три-файлах**.

Інші стандартні модулі входять до складу бібліотечного файла **TURBO.TPL**.

Модуль *System* підключається до кожної програми автоматично, *всі інші* стають доступними після підключення у розділі **Uses**.

Приклад побудови модуля користувача

Задача.

У заданому
тексті
підрахувати
кількість
слів, які
починаються
з літери 'а'.

Procedure
FindLetter

Program Strings1;

Uses Crt;

Var St, St1: string; n, p, k: byte;

BEGIN

clrscr; write (' Enter text: '); readln (St);

n:=length (St); k:=0;

St1:= Copy (St,1,1); p:=0;

repeat

if (St1='a') or (St1='A') then inc (k);

Delete (St,1,p); p:= Pos (' ',St);

St1:= Copy (St, p+1,1);

until p=0;

writeln (' K :', k:5); repeat until keypressed;

END.

локальні змінні

Procedure
EnterData



Приклад побудови модуля користувача

Текст укрупненої та вдосконаленої програми:

```
Program String2;
```

```
Uses Crt;
```

```
Var St, St1: string; L: char; k: byte;
```

```
Procedure EnterData (var St: string; var L:char);
```

```
begin
```

```
  clrscr;
```

```
  write ('  Enter text: '); readln (St);
```

```
  write ('  Enter small letter: '); readln (L);
```

```
end;
```



Приклад побудови модуля користувача

```
Procedure FindLetter (St: string; L:char; var k: byte);  
var n, p: byte;  
begin  
  n:=length (St);  
  k:=0;  
  St1:= Copy (St,1,1); p:=0;  
  repeat  
    if (St1=L) or (St1=UpCase(L)) then inc(k);  
    Delete (St,1,p);  
    p:= Pos (' ',St);  
    St1:= Copy (St, p+1,1);  
  until p=0;  
end;
```




Приклад побудови модуля користувача

BEGIN

EnterData (St, L);

FindLetter (St,L,k);

writeln;

**writeln (' Amout of the words which begin from the set
letter:', k:5);**

repeat until keypressed;

END.



Приклад побудови модуля користувача

Текст модуля:

Unit MyStr;

Interface

Var St, St1: string; k: byte; L: char;

Procedure EnterData (var St: string; var L:char);

Procedure FindLetter (St: string; L:char; var k: byte);



Приклад побудови модуля користувача

Implementation

Uses Crt;

Procedure EnterData;

begin

clrscr;

write (' Enter text: '); readln (St);

write (' Enter small letter: '); readln (L);

end;



Приклад побудови модуля користувача

Procedure FindLetter;

var n, p: byte;

begin

n:=length (St); k:=0; St1:= Copy (St,1,1); p:=0;

repeat

if (St1=L) or (St1=UpCase(L)) then inc(k);

Delete (St,1,p);

p:= Pos (' ',St);

St1:= Copy (St, p+1,1);

until p=0;

end;

BEGIN

END.



Приклад побудови модуля користувача

Текст основної програми:

```
Program Strings3;  
Uses Crt, MyStr;  
BEGIN  
EnterData (St, L);  
FindLetter (St,L,k);  
writeln;  
write(' Amout of the words which begin from the set  
letter:', k:5);  
repeat until keypressed;  
END.
```



Дякую за увагу!
