Тема:

Модульне програмування. Turbo Pascal

## План

1. Призначення модулів та їх структура 3

2. Компіляція програм, які використовують модулі 7

2.1 Стандартні модулі 9

Модуль Crt 10

Модуль Dos 11

Модуль Graph 12

Використана література 13

## 1. Призначення модулів та їх структура

Коли говорять про файли, які містять програми на Turbo Pascal, мова йде про три типи файлів.

а) текстові файли з текстами програм на Turbo Pascal (зазвичай ці файли мають розширення pas, хоча й це не обов’язково);

б) tpu - файли;

в) готових до виконання файлів, отриманих в результаті програм (exe - файли).

Що ж таке tpu - файли?

З розвитком обчислювальної техніки з’явилася можливість в масовому порядку вирішувати складні задачі, які вимагали написання великих за обсягом програм з залученням для цього колективу розробників. Так з’явилось поняття модульного програмування. Під цим терміном розуміють і розбиття програми на окремі фрагменти, і створення бібліотек фрагментів, з яких можливо сформувати програму (в Turbo Pascal вставки в текст програми здійснюються за рахунок директиви компіляції {SI имя\_включаемого\_файла}, і написання підпрограм, в тому числі й зовнішніх (ця можливість також є в Turbo Pascal). Можна також створювати незалежно зберігаючі і компільовані програмні одиниці, які мають інтерфейс, який дозволяє вик5ористовувати різні засоби цих програмних одиниць в розроблювальних програмах. Такі програмні одиниці в Turbo Pascal називають модулями. В результаті їх компілювання створюються файли з розширенням. tpu (tpu - файли). Використання модулів дозволяє створювати великі програми (ехе - файли), хоча розмір окремого tpu - файла не може перевищувати 64 Кбайт.

В Turbo Pascal модуль (unit) вважається окремою програмою і створюється спочатку як звичайний pas - файл, оформлений за певними правилами.

unit імя\_модуля; {Заголовок модуля}

interface {Заголовок інтерфейсної частини}

uses список\_використовуваних\_модулів

type …

const …

var …

procedure.

function …

implementation {Заголовок розділу реалізації}

{Опис локальних модулів, типів, констант, змінних, процедур і функцій, а також процедур і функцій, заголовки яких оголошені в розділі interface}

вegin

{Розділ ініціалізації}

end.

Як і програма, текст модуля розпочинається з заголовку і закінчується службовим словом еnd з крапкою. Але заголовок модуля розпочинається не зі слова program, а зі службового слова unit і є обов’язковим. Після слова unit вказується ім’я модуля, яке повинно співпадати з ім’ям файлу, який містить модуль (наприклад, для заголовка unit module; текст модуля повинен зберігатися в файлі module. pas, в результаті компілювання створюється файл module. tpu). Завершується заголовок крапкою з комою. Наведене в заголовку ім’я модуля вказують при його підключенню до програми або іншого модуля. Щоб підключити модуль до програми, необхідно відразу ж за заголовком програми (якщо він є) після службового слова uses вказати ім’я підключає мого модуля (uses module;). Службове слово uses в програмі може зустрітися лише один раз, тому для підключення декількох модулів їх імена перераховують через кому (uses module1,module2;).

За заголовком модуля йде його інтерфейсна частина, яка розпочинається зі службового слова interface. В ній перераховуються програмні ресурси (константи, типи, змінні, заголовки процедур і функції), які призначені для використання іншими модулями і програмами. Описані тут елементи називаються видимими (зовнішніми). Вимоги до написання тут такі самі, як і в Turbo Pascal, але для процедур і функцій вказують лише заголовки, причому директиви externa1, forward і assembler не ставляться.

Відмітимо також, що в розділі uses перераховуються імена модулів, які використовуються в даному випадку. Але вони не доступні програмам, які використовують даний модуль. Тому щоб отримати в програмі доступ до ресурсів модуля, його необхідно оголосити явно, а не через інший модуль.

Після інтерфейсної частини йде розділ реалізації, який розпочинається зі службового слова implementation. Тут оголошуються невідомі поза модулем внутрішні елементи (їх ще називають невидимими або схованими) - локальні змінні, константи, типи, процедури і функції, а також модулі, ресурси яких використовуються в середині даного модуля. Крім того, тут здійснюється реалізація тих процедур і функцій, заголовки яких описані в інтерфейс ній частині і які відомі поза модулем. Такий метод опису процедур і функцій здійснюється за наступною причиною: користувачу для звернення до підпрограми достатньо знати її ім’я і список параметрів. Знання способу її реалізації зовсім не потрібно. Якщо здійснюється модифікація підпрограми, яка розміщена в модулі, то достатньо змінити тільки її тіло в розділі реалізації, не змінюючи заголовка інтерфейсної частини. В зв’язку з цим програми, які використовують таку підпрограму, не змінюються. Якщо підпрограма оголошена в інтерфейс ній частині, то в розділі реалізації її заголовок зазвичай надається тільки у вигляді імені без вказування переліку і типів параметрів, а також типу повернення значення для функції.

Розділи іnterface та implementation обов’язкові, навіть якщо вони пусті, заголовки вказати необхідно.

Після розділу реалізації в модулі розташовується розділ ініціалізації, який розпочинається зі слова begin, за яким йдуть оператори, які будуть виконуватися до операторів з тіла програми (наприклад, встановлення початкових значень для деяких змінних, оголошених в модулі). Розділ ініціалізації не є обов’язковим, тому. Якщо при підключені модуля не потрібно робити ніяких початкових встановлень, він може не використовуватися (разом зі службовим словом begin).

## 2. Компіляція програм, які використовують модулі

До програми можна підключити декілька модулів. В свою чергу модуль може використовувати ресурси інших модулів. В цьому випадку в програмі в операторі uses вказують тільки імена тих модулів, які безпосередньо використовуються в програмі. Характерною особливістю модулів є те, що вони не повністю включаються в exe - файл: до програми додається лише те, що використовується. Заборонено звернення модуля до самого себе, хоча, якщо модуль оголошується в розділі реалізації, це обмеження розповсюджується лише на пряме звернення.

Можлива й така ситуація, коли в програмі і інтерфейс них частинах декількох модулів, підключених до неї, оголошені за допомогою одного і того самого імені різні об’єкти (наприклад, в програмі є розділ uses module1,module2; і як в програмі, так і в інтерфейс них частинах обох модулів існують описи з іменем name). В цьому випадку програма оперує передусім, елементами, які оголошені в ній самій. (тобто тим елементом name, який оголошений в програмі). Для звернення до об’єкту, ім’я якого перекрито, необхідно зліва до імені об’єкту дописати ім’я модуля, в якому він оголошений, об’єднав їх крапкою (наприклад, module1. name або System. Real).

В Turbo Pascal немає зв’язку між іменем програми і іменем файлу, в якому вона зберігається. Але для підключення модуля до програми його необхідно знайти на диску. Тому ім’я модуля і ім’я файлу, який містить модуль, повинні співпадати. Вихідний текст модуля зберігається в файлі з розширенням pas, а отриманий в результаті компіляції код модуля - в файлі з розширенням tpu (від Turbo Pascal Unit). Якщо ім’я модуля module, то відповідними файлами будуть module. pas і module. tpu.

При компіляції програми за допомогою команди Compile - > Compile компілятор послідовно відшуковує tpu - файли, які містять коди модулів які використовуються, для їх підключення. Пошук відбувається наступним чином:

1. Перегляд системного бібліотечного файлу модулів turbo. tp1 (від Turbo Pascal Library), який повинен зберігатися в тому самому каталозі, що й файл turbo. exe.

Якщо модуля немає в файлі turbo. tp1, то пошук продовжується в поточному каталозі.

Якщо і тут модуль не знайдений, то йде перегляд каталогів, перерахованих в розділі Unit Directories діалогового вікна, яке викликається командою меню Options - > Directories (перерахування каталогів відбувається через крапку з комою). Якщо модуль не виявлено, буде створене повідомлення про помилку і компіляція програми буде завершена.

При обрані команди компіляції Compile - >Make здійснюється пошук pas - файлу, який містить вихідний текст модуля. Виявивши його, в тому ж самому каталозі необхідно знайти відповідний tpu - файл. Якщо tpu - файл створений пізніше pas - файлу, то йде підключення модуля з tpu - файлу. Якщо ж tpu - файл не виявлений або pas - файл коректувався після створення tpu - файлу, то обов’язково компілюється нова версія tpu - файлу.

При компіляції за допомогою команди Compile - >Build обов’язково йде компіляція всіх pas - файлів, що містять тексти модулів, які використовуються. Якщо якогось з pas - файлів немає, але є відповідний tpu - файл, відбувається підключення останнього.

Модулі завжди компілюються перед компіляцією програми.

Системний бібліотечний файл модулів turbo. tp1 (в Turbo Pascal він єдиний) володіє спеціальною структурою, орієнтованої на швидкий пошук модулів, які в ньому містяться. Включення і виключення модулів виконується за допомогою спеціальної програми tpumover. exe. Пошук модулів в файлі turbo. tp1 відбувається завжди, тому не стоїть робити його дуже великим. Модулі, які часто використовуються, необхідно поміщати в файл turbo. tp1.

## 2.1 Стандартні модулі

В Turbo Pascal є вісім стандартних модулів (System, Dos, Crt, Printer, Overlay, Graph, Turbo3, Graph3). Модуль System підключається автоматично, і його не потрібно згадувати у розділі uses. Всі останні модулі необхідно оголошувати в програмі для забезпечення доступу до їх ресурсів.

В модулі System зберігаються типи, константи, змінні і підпрограми авторської версії мови, стандартної мови, а також ряд додаткових підпрограм. Відсутність цього модуля призводить до неможливості компіляції програми.

Модуль Crt використовується при роботі з екраном у текстовому режимі, під час читання з клавіатури, а також забезпечує деякі інші функції.

Модуль Dos забезпечує звернення до засобів операційної системи MS DOS.

Модуль Printer дозволяє здійснювати вивід на принтер за допомогою процедур Write і WriteLn. Для цього оголошена файлова змінна Lst типу Text, яка зв’язана з логічним пристроєм prn.

Модуль Graph забезпечує роботу з екраном у графічному режимі.

Модуль Overlay використовують для ефективної роботи великих програм. Модулі Turbo3 і Graph3 застосовують дуже рідко, їх використовують для забезпечення сумісності пізніших версій мови з програмами, які написані на мові Turbo Pascal 3.0.

Перші п’ять згадані модулі часто включають в файл turbo. tp1.

Є також цілий ряд інших модулів, які стали практично стандартні. Це модулі, які входять в бібліотеку Turbo Vision (Objects, Views, Dialogs, App і ін) і призначені для побудови інтерфейс них частин розроблювальних програм.

Для роботи з довгими (більше 255 символів) строками розроблений модуль Strings.

## Модуль Crt

Одним з найбільш часто використовуваних стандартних модулів є модуль Crt. В ньому є багато процедур і функцій для роботи з текстом. Серед них: засоби вводу, виведення і форматування тексту, створення і робота з вікнами та багато іншого.

Основним засобом відображення інформації, яка вводиться і виводиться під час роботи прикладних і системних програм є екран. В Turbo Pascal є можливість працювати з екраном у текстовому і графічному режимах.

Текстовий режим призначений для відображення символів кодової таблиці і характеризується максимальним числом символів в рядку і кількістю рядків на екрані. Крім того, монохромні дисплеї характеризуються кількістю степенем яскравості, а кольорові - можливою кількістю використовуваних кольорів. Мінімальною одиницею керування в текстових режимах є символ. Він складається з декількох пікселів, перетворення яких відбувається на апаратному рівні. Для зберігання символу, який виводиться на екран в текстовому режимі необхідно 2 байта. Перший байт містить безпосередньо символ кодової таблиці, а другий визначає, як символ має бути виведений на екран (його колір, колір фону, мерехтіння).

Засоби, які працюють з екраном в текстовому режимі зібрані в модулі Crt, який підключається звичайним способом:

uses Crt;

Встановлення текстового режиму відбувається за допомогою процедури TextMode, виконання якої призводить до очищення екрану і активації вказаного режиму. В таблиці запропоновані константи, які визначені в модулі Crt для встановлення текстових режимів.

Для встановлення того чи іншого текстового режиму необхідно звернутися до процедури TextMode:

TextMode (режим);

Константи, визначені для текстових режимів екрана

|  |  |
| --- | --- |
| Константа | Текстовий режим |
| BW40 | Розмір 40 стовпців на 25 рядків (40х25) для кольорового монітора (BW40=0)  |
| BW80 | Чорно-біле зображення з розміром екрана 80х25 (BW80=2)  |
| CO40 | Кольоровий текстовий режим 40х25 (СО40=1)  |
| CO80 | Кольоровий текстовий режим з розміром екрану 80х25 (СО80=3)  |
| Mono | Монохромний режим 80х25 |
| C40 i C80 | Константи СО40 і СО80 відповідно. Введені для сумісності з Turbo Pascal 3.0 |

Тут в якості параметра задається одна з перерахованих в таблиці констант. Наприклад, для встановлення кольорового режиму з розміром екрану 40х25 можна використати один з наступних варіантів:

TextMode(СО40); TextMode(С40); TextMode(3);

Процедура TextMode може бути використана без параметрів. В такому випадку активується режим, який зустрічався останнім до поточного режиму. За замовчуванням прийняті режими BW80 і С80.

В модулі Crt є константа Font8x8 зі значенням 256. Якщо додати її до основної константи, то кількість рядків буде збільшено до 43 для адаптерів EGA і 50 для адаптерів VGA. Наприклад:

TextMode(СО80+Font8x8);

## Модуль Dos

Існує цілий клас задач, рішення яких потребує застосування засобів операційної системи. Без цього не можливе написання практично майже ні однієї професійної програми.

В модулі Dos зосереджені функції і процедури, які забезпечують доступ до засобів MS DOS. При звернені до засобів операційної системи, зокрема при використані деяких процедур модуля Dos, можлива поява помилок. Для їх виявлення введена змінна DosError, в яку заноситься код помилки. Можливі значення цієї змінної перераховані в таблиці.

Коди помилок змінної DosError.

|  |  |
| --- | --- |
| Код помилки | Пояснення |
| 0 | Нормальне завершення |
| 2 | Файл не знайдений |
| 3 | Шлях не знайдений |
| 4 | Дуже багато відкритих файлів |
| 5 | Доступ до файлу закритий |
| 6 | Порушена інформація в полях файлу або системних областях |
| 8 | Недостатньо пам’яті |
| 10 | Несумісність параметрів оточення |
| 11 | Нерозпізнаний формат диску |
| 18 | Немає більше файлів |

Рекомендується здійснювати контроль коректності виконання процедур модуля Dos, зверненням до змінної DosError відразу ж після виклику процедур.

## Модуль Graph

Перші дисплеї (монітори) були монохромні. Їх представниками є монітори MDA (Mono Digitals Adapter) та Hercules, які зустрічаються зараз досить рідко. Із кольорових моніторів нещодавно були розповсюджені монітори GGA (Color Graphics Adapter) та EGA (Enhanced Graphics Adapter), зараз - VGA (Video Graphics Adapter) та SVGA (Super-VGA). Кольорове зображення отримуємо за рахунок підсвічення люмінофорних крапок трьох кольорів (красний, зелений, синій). При цьому кожна точка зображення (піксель) утворюється трьома поряд розташованими люмінофорними точками.

Монітори EGA i VGA фактично стали стандартом для тих додатків, які забезпечуються графічними можливостями мови Turbo Pascal. Особливістю моніторів VGA в тому, що в них велика роздільна здатність, і відстань між сусідніми пік селями по вертикалі і горизонталі співпадають.

В системному модулі Turbo Pascal не передбачені можливості по роботі в графічному режимі. Це передбачено в модулі Graph.

## Використана література

1. Безменов Н.И. Turbo Pascal 7.0. Руководство программиста. - М.: Изд-во Эксмо, 2006. - 160с.

2. Злобін Г.Г., Рикалюк Р. Є. Архітектура та апаратне забезпечення ПЕОМ: Навч. посіб. - К.: Каравела, 2006