

§19. THƯ VIỆN CHƯƠNG TRÌNH CON CHUẨN

Mỗi ngôn ngữ lập trình đều có một số lượng phong phú các chương trình con chuẩn trong các thư viện. Dưới đây giới thiệu sơ lược nội dung của một số thư viện chương trình con chuẩn của Pascal.

1. CRT

Thư viện *crt* chứa các thủ tục liên quan đến việc quản lí và khai thác màn hình, bàn phím của máy tính. Dùng các thủ tục của thư viện này, người lập trình có thể điều khiển việc đưa dữ liệu ra màn hình, xây dựng các giao diện màn hình-bàn phím, dùng bàn phím điều khiển chương trình hoặc sử dụng âm thanh để xây dựng các chương trình mô phỏng.

Ngoài thủ tục *clrscr* đã giới thiệu, dưới đây là một số thủ tục tiện ích khác:

Thủ tục *TextColor(color)* đặt màu cho chữ trên màn hình, trong đó *color* là hằng hoặc biến xác định màu và có thể nhận giá trị trong bảng sau:

Màu	Tên hằng	Giá trị
Đen	black	0
Xanh trời	blue	1
Xanh lá	green	2
Xanh lơ	cyan	3
Đỏ	red	4
Tím	magenta	5
Vàng	yellow	14
Trắng	white	15

Thủ tục *TextBackground(color)* đặt màu cho nền của màn hình, trong đó *color* là hằng hoặc biến xác định màu và có thể nhận giá trị trong bảng trên.

Thủ tục *GotoXY(x, y)* đưa con trỏ tới vị trí cột *x* dòng *y* của màn hình văn bản. Do màn hình văn bản gồm 25 dòng và 80 cột nên phạm vi giá trị của các tham số là $1 \leq x \leq 80, 1 \leq y \leq 25$.

2. GRAPH

Mặc dù Pascal không phải là ngôn ngữ chuyên về đồ hoạ, nhưng với thư viện đồ hoạ người lập trình có thể khai thác khả năng đồ hoạ của máy tính ở những mức độ thông dụng.

Thư viện này chứa các hàm, thủ tục liên quan đến chế độ đồ hoạ của các loại màn hình khác nhau và cho phép thực hiện các thao tác đồ hoạ cơ bản như vẽ điểm, đường, tô màu,...

a) Các thiết bị và chương trình hỗ trợ đồ hoạ

Màn hình có thể làm việc trong hai chế độ: *chế độ văn bản* và *chế độ đồ hoạ*. Có thể hình dung màn hình như một bảng các điểm sáng. Hình ảnh đồ hoạ được xây dựng từ các điểm sáng. Mỗi điểm sáng là một điểm ảnh (pixel) và điểm ảnh là đơn vị cơ sở của màn hình đồ hoạ.

Bảng mạch điều khiển màn hình là thiết bị đảm bảo tương tác giữa bộ xử lí và màn hình để thực hiện các chế độ phân giải và màu sắc. Tên gọi của bảng mạch điều khiển màn hình thường trùng với loại màn hình, ví dụ VGA, SVGA,...

Trong Turbo Pascal, thư viện *graph* cung cấp các chương trình điều khiển tương ứng với các loại bảng mạch đồ họa. Các chương trình điều khiển này nằm trong các tệp có phần mở rộng là BGI (Borland Graphics Interface). Muốn hoạt động trong chế độ đồ họa, cần phải có tệp BGI thích hợp với màn hình đang dùng. Các tệp BGI được Pascal ngầm định để trong thư mục con BGI. Khi khởi động đồ họa, cần chỉ rõ đường dẫn đến các tệp này.

Toạ độ trên màn hình đồ họa được đánh số từ 0, cột được tính từ trái sang phải và dòng được tính từ trên xuống dưới. Độ phân giải màn hình VGA thường được đặt là 640×480.

b) Khởi tạo chế độ đồ họa

Một chương trình đồ họa bao giờ cũng mở đầu bằng việc khởi tạo chế độ đồ họa. Các thủ tục, hàm của thư viện *graph* chỉ hoạt động khi chế độ đồ họa đã được thiết lập. Khi kết thúc làm việc với chế độ đồ họa thì cần quay về chế độ văn bản.

Thủ tục sau dùng để thiết lập chế độ đồ họa:

```
procedure InitGraph(var driver,mode:integer; path:string);
```

trong đó:

- *driver* là số hiệu của trình điều khiển BGI;
- *mode* là số hiệu của độ phân giải;
- *path* là đường dẫn đến các tệp BGI.

Thông thường, ta nên sử dụng cách thiết lập chế độ đồ họa tự động với biến *driver* được gán giá trị 0 (được định sẵn bằng hằng *detect*).

Ví dụ

Giả sử màn hình làm việc là VGA và các tệp BGI đang để ở thư mục C:\TP\BGI, khi đó các lệnh sau sẽ thiết lập đồ họa với chế độ VGAHi:

```
driver:= 0;  
InitGraph(driver, mode, 'C:\TP\BGI');
```

Sau khi kết thúc làm việc với chế độ đồ họa, để trở về chế độ văn bản ta gọi thực hiện thủ tục:

```
CloseGraph;
```

c) Các thủ tục vẽ điểm, đoạn thẳng

Vẽ điểm và đoạn thẳng là hai thao tác cơ bản của đồ hoạ.

- Trước khi vẽ, ta có thể đặt màu cho nét vẽ bằng thủ tục:

```
procedure SetColor(color: word);
```

- Vẽ điểm được thực hiện bằng thủ tục:

```
procedure PutPixel(x,y: integer; color: word);
```

trong đó:

– x và y là toạ độ của điểm;

– $color$ là màu của điểm.

- Đoạn thẳng được xác định bởi toạ độ của hai điểm đầu, cuối. Để vẽ đoạn thẳng nối hai điểm ta sử dụng thủ tục:

```
procedure Line(x1, y1, x2, y2: integer);
```

trong đó $(x1, y1)$ và $(x2, y2)$ là các toạ độ của hai điểm đầu, cuối của đoạn thẳng.

- Vẽ đoạn thẳng nối điểm hiện tại (vị trí con trỏ) với điểm có toạ độ (x, y) :

```
procedure LineTo(x, y: integer);
```

- Vẽ đoạn thẳng nối điểm hiện tại với điểm có toạ độ bằng toạ độ hiện tại cộng với gia số (dx, dy) :

```
procedure LineRel(dx, dy: integer);
```

d) Các thủ tục và hàm liên quan đến vị trí con trỏ

- Các hàm xác định giá trị lớn nhất có thể của toạ độ màn hình X và Y (để biết độ phân giải màn hình trong chế độ đồ hoạ đang sử dụng):

```
function GetMaxX: integer;
```

```
function GetMaxY: integer;
```

- Thủ tục chuyển con trỏ tới toạ độ (x, y) :

```
procedure MoveTo(x, y: integer);
```

e) Một số thủ tục vẽ hình đơn giản

- Vẽ đường tròn có tâm tại (x, y) , bán kính r :

```
procedure Circle(x,y: integer; r: word);
```

- Vẽ cung của elip có tâm tại điểm (x, y) với các bán kính trục Xr, Yr từ góc khởi đầu $StAngle$ đến góc kết thúc $EndAngle$:

```
procedure Ellipse(x, y: integer; StAngle, EndAngle, Xr, Yr: word);
```

- Vẽ hình chữ nhật có các cạnh song song với các trục toạ độ, $(x1, y1)$ là toạ độ của đỉnh trái trên còn $(x2, y2)$ là toạ độ của đỉnh phải dưới:

```
procedure Rectangle(x1, y1, x2, y2: integer);
```

3. Một số thư viện khác

System: Trong thư viện chuẩn này chứa các hàm sơ cấp và các thủ tục vào/ra mà các chương trình đều dùng tới.

Dos: Thư viện này chứa các thủ tục cho phép thực hiện trực tiếp các lệnh như tạo thư mục, thiết lập giờ hệ thống,...

Printer: Thư viện này cung cấp các thủ tục làm việc với máy in.

4. Sử dụng thư viện

Muốn sử dụng các thủ tục và hàm chuẩn của một (một số) thư viện nào đó (trừ *system*) phải dùng lệnh khai báo:

```
uses unit1, unit2, ..., unitN;
```

trong đó, *uses* là từ khoá, *unit1, unit2, ..., unitN* là tên các thư viện (được viết cách nhau bởi dấu phẩy).

Khai báo này phải là lệnh đầu tiên trong phần khai báo (nghĩa là nó chỉ viết sau khai báo tên chương trình).

Ví dụ

Để sử dụng các hàm và thủ tục chuẩn trong các thư viện *crt, dos, graph*, ta cần khai báo:

```
uses crt, dos, graph;
```