TRƯỜNG ĐẠI HỌC NHA TRANG

**KHOA CÔNG NGHỆ THÔNG TIN**

🙡✪🙣



ThS. NGUYỄN THỦY ĐOAN TRANG

TÀI LIỆU THAM KHẢO

**HỆ THỐNG THÔNG TIN ĐỊA LÝ**

**Khoa học máy tính**

**GIScience**

**Khoa học địa lý**

**Khoa học toán**

**NHA TRANG – 2018**

**MỤC LỤC**

[*CHƯƠNG 1* TỔNG QUAN VỀ HỆ THỐNG THÔNG TIN ĐỊA LÝ 4](#_Toc533409972)

[**1.1.** **BẢN ĐỒ** 4](#_Toc533409973)

[**1.2.** **CÁC ĐẶC TÍNH ĐỊA LÝ CỦA BẢN ĐỒ** 4](#_Toc533409974)

[**2.1.1.** **Bản đồ nền - Cơ sở địa lý của Bản đồ địa lý chuyên đề** 4](#_Toc533409975)

[**2.1.2.** **Ký hiệu bản đồ** 5](#_Toc533409976)

[**2.1.3.** **Phương pháp thể hiện nội dung bản đồ** 6](#_Toc533409977)

[**1.3.** **PHÉP CHIẾU BẢN ĐỒ** 6](#_Toc533409978)

[**1.4.** **HỆ QUI CHIẾU VN-2000** 12](#_Toc533409979)

[**1.4.1.** **Hệ tọa độ quy chiếu** 12](#_Toc533409980)

[**1.4.2.** **Hệ độ cao** 13](#_Toc533409981)

[**1.5.** **KHÁI NIỆM HỆ THỐNG THÔNG TIN ĐỊA LÝ** 13](#_Toc533409982)

[**1.5.1.** **Thông tin địa lý** 13](#_Toc533409983)

[**1.5.2.** **Hệ thống thông tin địa lý** 14](#_Toc533409984)

[**1.5.3.** **Khoa học thông tin địa lý** 15](#_Toc533409985)

[**1.6.** **TIẾN TRÌNH PHÁT TRIỂN GIS** 16](#_Toc533409986)

[**1.7.** **CHỨC NĂNG CỦA GIS** 17](#_Toc533409987)

[**1.7.1.** **Thu thập dữ liệu** 17](#_Toc533409988)

[**1.7.2.** **Lưu trữ dữ liệu** 18](#_Toc533409989)

[**1.7.3.** **Phân tích dữ liệu** 19](#_Toc533409990)

[**1.7.4.** **Hiển thị dữ liệu** 20](#_Toc533409991)

[**1.8.** **CÁC THÀNH PHẦN CỦA HỆ THỐNG THÔNG TIN ĐỊA LÝ** 20](#_Toc533409992)

[**1.8.1.** **Thành phần dữ liệu** 20](#_Toc533409993)

[**1.8.2.** **Phần cứng** 21](#_Toc533409994)

[**1.8.3.** **Phần mềm** 22](#_Toc533409995)

[**1.8.4.** **Con người** 22](#_Toc533409996)

[**1.9.** **CÁC LĨNH VỰC ỨNG DỤNG GIS** 23](#_Toc533409997)

[**1.10.** **KHUYNH HƯỚNG PHÁT TRIỂN GIS** 23](#_Toc533409998)

[**1.10.1.** **Khuynh hướng phát triển về lý thuyết** 23](#_Toc533409999)

[**1.10.2.** **Khuynh hướng phát triển phần mềm** 23](#_Toc533410000)

[**1.10.3.** **Khuynh hướng phát triển ứng dụng** 24](#_Toc533410001)

[*CHƯƠNG 2* MÔ HÌNH DỮ LIỆU CỦA HỆ THỐNG THÔNG TIN ĐỊA LÝ 25](#_Toc533410002)

[**2.1.** **CẤU TRÚC DỮ LIỆU RASTER** 25](#_Toc533410003)

[**2.1.1.** **Dạng hình học và topology của dữ liệu Raster** 26](#_Toc533410004)

[**2.1.2.** **Các thuật toán nén dữ liệu cấu trúc raster** 27](#_Toc533410005)

[**2.2.** **CẤU TRÚC DỮ LIỆU VECTOR** 29](#_Toc533410006)

[**2.2.1.** **Tính hình học của cấu trúc vector** 29](#_Toc533410007)

[**2.2.2.** **Cấu trúc Spaghetti** 29](#_Toc533410008)

[**2.2.3.** **Cấu trúc topology** 30](#_Toc533410009)

[**2.2.4.** **Những đặc điểm topology của cấu trúc vector** 32](#_Toc533410010)

[**2.3.** **MÔ HÌNH DỮ LIỆU MẠNG** 36](#_Toc533410011)

[**2.4.** **MÔ HÌNH DỮ LIỆU TIN** 37](#_Toc533410012)

[**2.5.** **DỮ LIỆU THÔNG TIN QUẢN LÝ DỮ LIỆU (METADATA)** 38](#_Toc533410013)

[**2.6.** **ƯU NHƯỢC ĐIỂM CỦA CẤU TRÚC DỮ LIỆU RASTER VÀ VECTOR** 38](#_Toc533410014)

[*CHƯƠNG 3* THU THẬP THIẾT KẾ VÀ XÂY DỰNG DỮ LIỆU GIS 40](#_Toc533410015)

[**3.1.** **THU THẬP DỮ LIỆU KHÔNG GIAN** 40](#_Toc533410016)

[**3.1.1.** **Thu thập dữ liệu** 40](#_Toc533410017)

[**3.1.2.** **So sánh các nguồn thu thập dữ liệu** 41](#_Toc533410018)

[**3.2.** **THIẾT KẾ DỮ LIỆU KHÔNG GIAN** 41](#_Toc533410019)

[**3.2.1.** **Dữ liệu không gian** 42](#_Toc533410020)

[**3.2.2.** **Dữ liệu thuộc tính** 42](#_Toc533410021)

[**3.3.** **SỐ HÓA BẢN ĐỒ** 43](#_Toc533410022)

[**3.3.1.** **Tỉ lệ bản đồ** 44](#_Toc533410023)

[**3.3.2.** **Chọn lưới chiếu bản đồ** 44](#_Toc533410024)

[**3.3.3.** **Nhập dữ liệu bản đồ** 44](#_Toc533410025)

[*CHƯƠNG 5* TRUY VẤN PHÂN TÍCH VÀ HIỂN THỊ DỮ LIỆU 46](#_Toc533410029)

[**4.1.** **TRUY VẤN TỪ DỮ LIỆU THUỘC TÍNH** 47](#_Toc533410030)

[**4.2.** **TRUY VẤN TỪ DỮ LIỆU KHÔNG GIAN** 48](#_Toc533410031)

[**4.3.** **PHÂN TÍCH KHÔNG GIAN** 51](#_Toc533410032)

[**4.4.** **KÝ HIỆU HÓA VÀ TẠO TRANG IN BẢN ĐỒ** 51](#_Toc533410033)

[**4.4.1.** **Ký hiệu hóa** 51](#_Toc533410034)

[**4.4.2.** **Tạo trang in bản đồ** 54](#_Toc533410035)

[*CHƯƠNG 5* LẬP TRÌNH GIS CƠ BẢN VỚI NGÔN NGỮ AVENUE 57](#_Toc533410038)

[**5.1.** **CƠ BẢN VỀ NGÔN NGỮ LẬP TRÌNH AVENUE** 57](#_Toc533410039)

[**5.1.1.** **Giới thiệu ngôn ngữ Avenue** 57](#_Toc533410040)

[**5.1.2.** **Các đặc điểm của ngôn ngữ Avenue** 57](#_Toc533410041)

[**5.1.3.** **Giao diện lập trình Avenue** 58](#_Toc533410042)

[**5.1.4.** **Lớp và đối tượng** 58](#_Toc533410043)

[**5.1.5.** **Yêu cầu và cách gọi yêu cầu trong Avenue** 59](#_Toc533410044)

[**5.1.6.** **Các kiểu dữ liệu trong Avenue** 61](#_Toc533410045)

[**5.1.7.** **Các câu lệnh** 64](#_Toc533410046)

[**5.1.9.** **Đối tượng av và từ khoá self** 67](#_Toc533410052)

[**5.2.** **THAY ĐỔI GIAO DIỆN ARCVIEW** 70](#_Toc533410053)

[**5.2.1.** **Tạo Menu** 71](#_Toc533410054)

[**5.2.2.** **Xóa Menu** 73](#_Toc533410055)

[**5.3.** **KHUNG NHÌN (VIEW) VÀ LỚP CHỦ ĐỀ (THEME)** 73](#_Toc533410056)

[**5.3.1.** **Lược đồ mô hình đối tượng của View và Theme** 73](#_Toc533410057)

[**5.3.2.** **Tạo mới một khung nhìn** 73](#_Toc533410058)

[**5.3.3.** **Tìm kiếm một khung nhìn** 74](#_Toc533410059)

[**5.3.4.** **Gán đơn vị bản đồ và đơn vị thước đo cho khung nhìn** 74](#_Toc533410060)

[**5.3.5.** **Thêm lớp chủ đề (theme) vào khung nhìn** 75](#_Toc533410061)

[**5.3.6.** **Tìm một lớp chủ đề trong khung nhìn** 76](#_Toc533410062)

[**5.3.7.** **Thay đổi thứ tự của lớp chủ đề trong bảng nội dung** 77](#_Toc533410063)

[**5.3.8.** **Thay đổi biểu tượng của lớp chủ đề** 77](#_Toc533410064)

[**5.3.9.** **Xoá các lớp chủ đề khỏi khung nhìn** 78](#_Toc533410065)

[**5.3.10.** **Chọn đối tượng không gian của lớp chủ đề bằng chuột** 79](#_Toc533410066)

[**5.3.11.** **Chọn đối tượng không gian của lớp chủ đề bằng các hình đồ họa** 80](#_Toc533410067)

[**5.3.12.** **Chọn đối tượng không gian lớp chủ đề bằng một lớp chủ đề khác** 81](#_Toc533410068)

# ***CHƯƠNG 1*** TỔNG QUAN VỀ HỆ THỐNG THÔNG TIN ĐỊA LÝ

## **BẢN ĐỒ**

Bản đồ địa lý hiện nay được phân làm hai nhóm đó là bản đồ địa lý chung và bản đồ địa lý chuyên đề (hay gọi tắt là bản đồ chuyên đề).

Bản đồ địa lý chung là bản đồ thể hiện mọi đối tượng hiện tượng địa lý của bề mặt trái đất, bao gồm đầy đủ các đối tượng và hiện tượng tự nhiên, kinh tế, xã hội như thủy văn, địa hình, thực vật, đất đai, dân cư, giao thông, công nghiệp, nông nghiệp, lâm nghiệp, văn hóa, chính trị và hành chính.

Tùy thuộc tỉ lệ bản đồ mà mức độ nội dung của bản đồ địa lý chung có thể có nhiều chi tiết hoặc ít chi tiết hơn, nhưng về nguyên tắc với một tỷ lệ xác định thì bản đồ địa lý chung thể hiện đối tượng, hiện tượng với cùng mức độ chi tiết, nghĩa là không quá chú trọng đến một yếu tố nhất định.

Bản đồ địa lý chung được phân thành ba nhóm: bản đồ địa hình, bản đồ địa hình khái quát và bản đồ khái quát. Bản đồ địa hình được thành lập bằng phương pháp đo vẽ trực tiếp ngoài thực địa, có kết hợp với không ảnh. Đây là bản đồ có nội dung chi tiết và độ chính xác cao, được thành lập theo quy trình quy phạm nhà nước ở các tỷ lệ từ 1/100.000 và lớn hơn. Bản đồ địa hình khái quát và bản đồ khái quát là bản đồ có tỷ lệ 1/200.000, 1/500.000, 1/1.000.000 và nhỏ hơn, được thành lập từ bản đồ địa hình sẵn có với tỷ lệ lớn hơn.

Bản đồ địa lý chuyên đề chỉ thể hiện một vài yếu tố của bản đồ địa lý chung hoặc một vài hiện tượng, quá trình địa lý mà không thể hiện trên bản đồ địa lý chung. Bản đồ địa lý chuyên đề về một yếu tố nào đó sẽ đề cập đầy đủ các khía cạnh của yếu tố đó. Ví dụ khi trình bày về vấn đề dân cư cần phản ánh dân số, mật độ, thành phần xã hội, nghề nghiệp, dân tộc, độ tuổi… hay khi yếu tố khí hậu không có trên bản đồ địa lý chung nhưng trên bản đồ khí hậu chuyên đề thì lại được đề cập đầy đủ và hệ thống.

Bản đồ số là một tập có tổ chức các dữ liệu bản đồ trên thiết bị có khả năng đọc được bằng máy tính và được thể hiện dưới dạng hình ảnh bản đồ.

## **CÁC ĐẶC TÍNH ĐỊA LÝ CỦA BẢN ĐỒ**

### **Bản đồ nền - Cơ sở địa lý của Bản đồ địa lý chuyên đề**

Nói đến bản đồ địa lý chuyên đề là nói đến nội dung chuyên đề và cơ sở địa lý mà nội dung chuyên đề được thể hiện trên nó. Cơ sở địa lý là nền của bản đồ chuyên đề nên còn được gọi là bản đồ nền. Nếu chỉ có duy nhất nội dung chuyên đề thì tự nó không tạo ra bản đồ chuyên đề, lý do vì bản đồ chuyên đề thể hiện nội dung chọn lọc trong mối quan hệ tương hỗ với các yếu tố khác của cảnh quan môi trường địa lý. Chính vì vậy việc thành lập bản đồ nền là một trong những giai đoạn quan trọng của quá trình thành lập bản đồ chuyên đề.

### **Ký hiệu bản đồ**

Ký hiệu bản đồ là ngôn ngữ của bản đồ, là tập hợp các hình dạng (shape), đường nét (line), ký hiệu (symbol), đồ thị (chart), màu sắc (color) và các ghi chú (legend) để phản ảnh các đối tượng, hiện tượng và truyền thông tin của chúng trong không gian. Ký hiệu bản đồ là ngôn ngữ khoa học nên phải có chức năng sau:

* Ký hiệu phải gợi sự liên tưởng đến đối tượng.
* Ký hiệu phải chứa đựng một nội dung nào đó về số lượng, chất lượng, cấu trúc và động lực phát triển của đối tượng.
* Ký hiệu phải phản ánh được vị trí và sự phân bố đối tượng trên bản đồ. Ngoài vị trí của đối tượng, còn có vị trí tương quan vừa phản ảnh quy luật phân bố hiện tượng vừa cho thấy đặc điểm phân bố của hiện tượng như phân bố liên tục, không liên tục, phân bố theo tuyến, theo diện tích hay phân bố rời rạc.

Để chọn ký hiệu phù hợp, cần biết đặc tính của đối tượng và hiện tượng. Các hiện tượng trong thực tế có thể phân bố theo các cách khác nhau như phân bố liên tục, không liên tục, phân bố lẻ tẻ. Hiện tượng phân bố liên tục là hiện tượng có ở mọi nơi trong không gian địa lý. Có những hiện tượng phân bố liên tục nhưng phân bố theo tuyến như sông ngòi, đường giao thông, đường ranh giới… Hiện tượng phân bố không liên tục là hiện tượng xuất hiện rời rạc.

Phương tiện thường dùng để tạo ra ký hiệu bản đồ là các dạng đồ họa, màu sắc, ký tự và số. Ký hiệu được vẽ theo quy ước nên được gọi là ký hiệu quy ước. Chúng được phân thành ba loại chính: ký hiệu theo tỷ lệ, ký hiệu ngoài tỷ lệ, ký hiệu nửa tỷ lệ.

Ký hiệu theo tỷ lệ (ký hiệu diện tích) dùng để biểu thị sự phân bố đối tượng trên diện tích lãnh thổ như rừng, đầm lầy, ao hồ, diện tích trồng trọt,… Chúng được thể hiện theo tỷ lệ bản đồ và giới hạn bằng ranh giới vùng phân bố, bên trong vùng phân bố có thể vẽ thêm các ký hiệu ngoài tỷ lệ.

Ký hiệu ngoài tỷ lệ (ký hiệu điểm) được vẽ theo kích thước quy định cho từng tỷ lệ hay nhóm tỷ lệ. Được sử dụng để biểu thị các đối tượng mà diện tích quá nhỏ, không thể thể hiện lên bản đồ theo tỷ lệ được như các điểm tam giác trắc địa trạm thủy điện, trạm khí tượng,…

Ký hiệu nửa tỷ lệ (ký hiệu dạng tuyến) được dùng để biểu thị các đối tượng có dạng hình tuyến như đường sá, sông ngòi, ranh giới,… Ký hiệu này chỉ thể hiện chiều dài hình tuyến theo tỷ lệ bản đồ còn chiều rộng đối tượng không thể hiện theo tỷ lệ bản đồ, nhất là những bản đồ tỷ lệ nhỏ. Nếu muốn thể hiện sông ngòi, đường sá theo tỷ lệ thì bản đồ phải có tỷ lệ cực lớn: 1/500; 1/200; 1/100. Khi đó cả chiều dài và chiều rộng được thể hiện hoàn toàn theo tỷ lệ bản đồ.

Nhờ khả năng phân hóa các ký hiệu, theo hình dạng, kích thước, phương hướng, màu sắc và cường độ màu sắc, theo cấu trúc bên trong của ký hiệu và khả năng phối hợp mà hệ thống ký hiệu có những khả năng đa dạng hơn.

### **Phương pháp thể hiện nội dung bản đồ**

Phương pháp thể hiện nội dung bản đồ có thể được phân loại theo định tính, định lượng hoặc phân loại theo khả năng định vị như định vị theo điểm, định vị theo diện tích, định vị toàn năng. Nội dung bản đồ có thể được thể hiện bằng phương pháp ký hiệu, phương pháp biểu đồ định vị, phương pháp đường chuyển động, đường đồng mức, phương pháp chấm điểm, phương pháp nền chất lượng, phương pháp bản đồ biểu đồ, phương pháp đồ giải.

Phương pháp ký hiệu là phương pháp thể hiện các đối tượng định vị tại những vị trí xác định trên bản đồ như trường học, khách sạn, khu dân cư,… gồm ký hiệu chữ, ký hiệu hình học, ký hiệu tượng trưng, ký hiệu hình vẽ. Mỗi ký hiệu gồm ba thành phần: hình dạng, kích thước và màu sắc. Hình dạng và màu sắc nêu lên đặc tính chất lượng, kích thước phản ánh định lượng của hiện tượng. Nếu tại một vị trí có nhiều đối tượng, việc sắp xếp các ký hiệu và ghi chú sẽ gặp nhiều khó khăn. Nếu các đối tượng đó đồng nhất và có thể so sánh được với nhau, có thể sử dụng một ký hiệu chung. Kích thước ký hiệu sẽ là tổng giá trị của các đối tượng. Các phần trong ký hiệu sẽ tương ứng với từng đối tượng. Trong các loại ký hiệu, ký hiệu hình học có thể coi là hiệu quả nhất, lý do chúng có hình dạng đơn giản, định vị chính xác, dễ vẽ, kích thước của ký hiệu thể hiện đặc tính số lượng.

## **PHÉP CHIẾU BẢN ĐỒ**

#### Khái quát phép chiếu bản đồ

Khi xem trái đất như một mặt cầu hoặc một mặt ellipsoid, để xây dựng một bản đồ cần chuyển đổi bề mặt ba chiều sang mặt phẳng hai chiều. Các thuật toán chuyển đổi này thường được gọi là phép chiếu bản đồ sử dụng các công thức toán học liên quan giữa hệ tọa độ cầu và hệ tọa độ phẳng.

Phép chiếu bản đồ là một hàm số thể hiện mối tương quan giữa một đối tượng trên thực tế và hình ảnh của nó trên bản đồ

**Chiếu mp**

A(ϕ,λ)

A’(x,y)

Phương trình chung của lưới chiếu:

*x = f1 (ϕ,λ)*

*y = f2 (ϕ,λ)*

Trong đó:

* x, y: tọa độ mặt phẳng
* ϕ, λ: tọa độ địa lý
* f1, f2: hàm

***Hình 2.7****. Biểu diễn đối tượng trên bản đồ.*

Mặt phẳng bản đồ.

1

2

3

1

Chiếu thẳng góc mặt toán học

2

Thu nhỏ theo tỉ lệ bản đồ cần vẽ

3

Chiếu xuống mặt phẳng bản đồ

Mặt toán học biểu diễn trái đất (Ellipsesoid)

Mặt toán học biểu điễn trái đất thu nhỏ theo tỷ lệ

Hệ tọa độ chiếu được định nghĩa trên mặt phẳng hai chiều. Khác với hệ tọa độ địa lý, hệ tọa độ chiếu có chiều dài, góc và diện tích cố định theo hai chiều được xây dựng trên cơ sở một hệ tọa độ địa lý xác định bởi một mặt cầu. Trong hệ tọa độ chiếu, vị trí được xác định bởi cặp tọa độ x, y.

Có nhiều phương pháp chiếu như phép chiếu hình trụ, hình nón, … ở nước ta thường sử dụng phép chiếu hình trụ ngang.

a) Phép chiếu hình trụ đứng

b) Phép chiếu hình trụ ngang

***Hình 2.8.*** *Những dạng thường gặp trong phép chiếu hình trụ*

#### Phân loại lưới chiếu bản đồ

Theo đặc điểm sai số: khi biểu diễn từ mặt ellipsoid hay mặt cầu lên mặt phẳng luôn có sai số, có 3 loại sai số cơ bản là sai số chiều dài, góc và diện tích. Tuy nhiên, các loại sai số cũng có liên quan chặt chẽ với nhau. Nghĩa là, khi giảm nhỏ sai số diện tích sẽ làm tăng sai số góc và ngược lại. Theo đặc điểm sai số của phép chiếu, người ta chia ra các loại phép chiếu sau: giữ góc, giữ diện tích và phép chiếu tự do (có sự tham gia của 3 loại sai số, tiêu biểu cho lưới chiếu tự do là lưới chiếu đồng khoảng cách).

* Theo vị trí mặt chiếu so với mặt cầu: lưới chiếu đứng, ngang, nghiêng
* Theo đặc điểm dạng lưới kinh tuyến, vĩ tuyến: lưới chiếu hình trụ, nón.

Sai số trên bản đồ có liên quan đến tỉ lệ, có 2 dạng tỉ lệ là tỉ lệ chung và tỉ lệ riêng. Trong đó:

* Tỉ lệ chung là tỉ số thể hiện mức độ thu nhỏ của bề mặt trái đất, không có sai số và tại bất kỳ điểm nào trên bản đồ đều có tỉ lệ chung giống nhau (1: 10000, 1: 100000).
* Tỉ lệ riêng là tỉ số chỉ sự phóng to hay thu nhỏ khi chiếu từ mặt ellipsoid thu nhỏ lên mặt phẳng, đặc trưng cho sự sai số của phép chiếu, tỉ lệ riêng tại mỗi điểm phân biệt là không giống nhau.

Mô hình phép chiếu và tỉ lệ có thể biểu diễn qua hình sau:

tỉ lệ chung

tỉ lệ riêng

*Bề mặt trái đất*

*Bề mặt Ellipsesoid*

*thu nhỏ*

*Mặt phẳng bản đồ*

thu nhỏ

chiếu

***Hình 2.9.***  *Mô hình phép chiếu và tỉ lệ*

#### Lưới chiếu bản đồ thường dùng

Bề mặt gồ ghề của trái đất là bề mặt phức tạp và không thể biểu diễn chính xác bởi một phương trình toán học. Muốn biểu diễn bề mặt trái đất lên bản đồ, chúng ta phải chiếu theo hướng thẳng góc tất cả các yếu tố bao gồm điểm, đường nét, diện tích, của nó lên một nền chung. Nền đó chính là bề mặt toán học đại diện một cách gần đúng nhất bề mặt thật của Trái Đất. Bề mặt này có thể là mặt phẳng, ellipsoid hoặc hình cầu và được chọn sao cho phù hợp nhất với hình dạng tự nhiên của Trái đất, đồng thời lại trùng khớp với mặt nước biển trung bình (mặt Geoid) nhiều nhất.

Chọn lựa loại bề mặt toán học nào là tuỳ thuộc vào diện tích khu vực cần vẽ bản đồ, và yêu cầu về độ chính xác của bản đồ cần vẽ. Trong trường hợp chiếu một lãnh thổ rộng lớn cần phải xem xét tới độ cong của bề mặt Trái Đất, ngay cả khi không yêu cầu độ chính xác cao.

Lưới bản đồ được thành lập trên cơ sở phép chiếu bản đồ. Phương pháp chiếu hình kinh tuyến và vĩ tuyến lên mặt phẳng gọi là phép chiếu bản đồ. Phép chiếu phải tìm ra được hệ thức toán học giữa mặt chiếu của trái đất và mặt mặt phẳng biểu diễn, tức là giữa toạ độ địa lý của mặt toán học đại diện cho trái đất và toạ độ phẳng vuông góc trên bản đồ. Vì thế vấn đề cơ bản của phép chiếu hình lưới kinh tuyến và vĩ tuyến của trái đất lên mặt phẳng bản đồ là xác định một phép chiếu phù hợp.

Lưới chiếu bản đồ là biểu hiện của lưới kinh tuyến, vĩ tuyến của mặt toán học đại diện cho Trái đất lên bản đồ theo một phép chiếu nào đó. Dựa vào lưới bản đồ người ta có thể xác định được vị trí của một điểm bất kỳ trên bản đồ.

Phép chiếu hình trụ được sử dụng phổ biến trong việc xây dựng các bản đồ địa hình tỉ lệ trung bình và tỉ lệ lớn. Phép chiếu hình trụ có thể có trục thẳng đứng hoặc trục nằm ngang. Phép chiếu UTM (Universal Transverse Mercator) thực chất là phép chiếu Gauss.

#### Phép chiếu hình trụ ngang đồng góc Gauss –Kruger

Đầu thế kỷ XIX, nhà toán học người Đức K.F.Gauss (30/04/1777 – 23/02/1855) đã công bố lí thuyết về phép chiếu đồng góc sau đó giáo sư L.Kruger (21/09/1857 – 01/06/1923) đã cụ thể hoá và tìm ra công thức cho cho phép chiếu hình trụ ngang đồng góc vào năm 1912. Vì vậy, phép chiếu này có tên gọi là Gauss-Kruger (đôi khi còn gọi là phép chiếu Gauss). Phép chiếu này được sử dụng ở nước ta để tính toạ độ phẳng cho các điểm khống chế trắc địa và chia mảnh hệ thống bản đồ cơ bản nhà nước.

Bản chất của phép chiếu Gauss-Kruger là xác định hệ thức toán học của mối liên hệ giữa toạ độ địa lý của các điểm trên bề mặt Ellipsoid và toạ độ vuông góc phẳng tương ứng trên mặt phẳng bản đồ, vì thế toạ độ vuông góc trên bản đồ cơ bản nhà nước Việt Nam còn gọi là toạ độ Gauss-Kruger.

Nếu đặt hình trụ nằm ngang thì trục của hình trụ nằm trong mặt phẳng xích đạo, đi qua tâm của Ellipsoid trái đất và tiếp xúc với mặt này theo một đường kinh tuyến nào đó. Đường kinh tuyến này không bị sai số trong phép chiếu, các kinh tuyến nằm cách kinh tuyến tiếp xúc về phía đông và phía tây đều bị sai số khi chiếu lên mặt phẳng bản đồ.

***Hình 2.10****. Phép chiếu hình trụ ngang*

P

P’

E

E’

Các vị trí nằm càng xa kinh tuyến trung tâm thì sai số càng lớn dần, nhằm giảm sai số theo yêu cầu về độ chính xác người ta chia mặt Ellipsoid trái đất ra làm nhiều múi kinh tuyến. Bề mặt Ellipsoid trái đất được chia làm cả thảy 60 múi nhị giác cầu, tương tự ranh giới chia mảnh bản đồ quốc tế tỉ lệ 1:1000 000. Sau khi căng mặt phẳng thì mỗi nhị giác cầu trở thành nhị giác phẳng được giới hạn bởi 2 cung kinh tuyến biên.

Kinh tuyến tiếp tuyến là kinh tuyến trung tâm của mỗi múi không có sai số độ dài đồng thời là trục tọa độ X của nhị giác phẳng còn được gọi là kinh tuyến trung tâm của múi. Xích đạo trong mỗi múi cũng được biểu thị là một đường thẳng, thẳng góc với trục X, và được chọn làm trục Y. Hai đường thẳng đó là trục toạ độ vuông góc mặt phẳng cho toàn múi (toàn nhị giác phẳng).

Trong phép chiếu này mỗi múi có một hệ thống toạ độ riêng, được biểu thị trực tiếp lên mặt phẳng bảo toàn góc chiếu và thực tế là đồng diện tích nữa. Ở mỗi múi các kinh tuyến khác có dạng cung hội tụ ở cực. Các vĩ tuyến là những đường cắt vuông góc với kinh tuyến. Độ cong của các kinh tuyến và vĩ tuyến ở mỗi mảnh bản đồ cơ bản là không đáng kể. Nước ta ở vùng vĩ độ thấp nên độ cong chỉ bắt đầu nhận thấy trên tờ bản đồ cơ bản tỉ lệ 1/200.000 và nhỏ hơn. Ở tất cả các mảnh bản đồ tỉ lệ lớn hơn 1/200.000 các đường cong kinh tuyến và vĩ tuyến đều là đường thẳng. Trong cùng một múi tất cả các mảnh bản đồ cùng tỉ lệ hoàn toàn có thể ghép lại với nhau mà không bị đứt quãng hay chồng chéo.

Kinh tuyến gốc

Kinh tuyến giữa

Xích đạo

30

90

1

2

3

0

60

120

***Hình 2.11****. Múi chiếu trong phép chiếu Gauss*

Khi Ellipsoid có dạng hình cầu, các sai số được tính như sau: So với múi trên Ellipsoid thì múi trên mặt phẳng bản đồ có phần rộng hơn. Vì vậy tỉ số độ dài trên bản đồ và trên Ellipsoid là:



Qua đó, sai lệch giữa độ dài trên bản đồ và trên Ellipsoid là:



Và sai số tương đối được tính như sau:



Trong đó:

* S là đoạn thẳng trên mặt bản đồ.
* s là đoạn thẳng tương ứng của S trên trái đất.
* y là khoảng cách từ điểm tính đến kinh tuyến giữa.
* R là bán kính trái đất.

Trong trường hợp Ellipsoid nói chung, công thức tính phức tạp hơn. Nếu dùng các công thức trên để tính sai số tương đối cho các giá trị s khác nhau ta có:

* Khi y=50 km thì sai số độ dài  bằng 1:32.000
* Khi y=100 km thì sai số độ dài  bằng 1:8.000
* Khi y=300 km thì sai số độ dài  bằng 1:890

Chiều rộng mỗi múi tại xích đạo là 668 km cho múi 6 độ hoặc là 334 km ở múi 3 độ. Sai số tương đối như trên tại vùng biên của mỗi múi thuộc khu vực xích đạo khoảng 1:800, sai số này không vượt quá giới hạn sai đo vẽ địa hình. Mỗi múi 6 độ được đánh số Ả rập từ 1 đến 60, bắt đầu đánh múi số 1 từ múi giới hạn bởi kinh tuyến 180 độ và kinh tuyến 186 độ về phía đông.

Lãnh thổ nước ta nằm trong phạm vi 4 múi: múi thứ 47, 48, 49, và 50 với các kinh tuyến trung tâm là 990, 1050, 1110, và 1170. Phần đất liền hầu hết nằm ở múi thứ 48 và 49. Toạ độ lưới km trên bản đồ theo phép chiếu Gauss được ghi lại trên khung bản đồ địa hình. Hệ thống lưới toạ độ vuông góc, còn gọi là toạ độ ô vuông, lưới km trong mỗi múi tính từ kinh tuyến giữa về phía đông được vẽ trên bản đồ không chỉ hạn chế đến độ rộng là 3 độ (trong múi 6 độ) mà đến 3030’ để có độ phủ là 30’ ở vùng ven của múi kề.

#### Phép chiếu UTM

Phép chiếu này thường sử dụng cho các bản đồ địa hình tỷ lệ 1:50.000, 1:25.000, 1:10.000 tại các tỉnh phía Nam do cục bản đồ của quân đội Mỹ sản xuất trước 1975. Phép chiếu UTM cũng là phép chiếu hình trụ ngang đồng góc nhưng hình trụ chiếu không tiếp xúc với mặt Ellipsoid tại kinh tuyến trung tâm mà cắt mặt Ellipsoid theo hai cát tuyến cách đều kinh tuyến trung tâm 180 km về 2 phía, khi đó tỉ lệ chiều dài không đổi trên 2 vòng cát tuyến, còn tỉ lệ chiếu dài trên kinh tuyến trục m=0,9996. Theo cách chiếu như vậy sẽ giảm được sai số biến dạng tại biên các múi chiếu và phân bố đều trong phạm vi múi chiếu 60. Đây cũng là ưu điểm của phép chiếu UTM so với phép chiếu Gauss.

Trong hệ toạ độ phẳng UTM cũng quy ước chuyển trục x về bên trái cách kinh tuyến trục 500 km. Còn trị số toạ độ x của gốc toạ độ được quy định khác nhau cho Bắc và Nam bán cầu. Trị số này ở Bắc bán cầu vẫn là 0 như hệ toạ độ Gauss, ở Nam bán cầu là 10.000 km. Một mảnh bản đồ thuộc Nam bán cầu, gốc toạ độ 0 được dịch chuyển sang phía Tây là 500 km về phía Nam là 10.000 km. Lúc đó gốc toạ độ có trị số Y=500 km, X=10.000 km. Nước ta nằm ở bắc bán cầu nên toạ độ của gốc toạ độ sẽ là Y=500 km và X=0.

180km

180km

Kinh tuyến giữa

0km

500km

Kinh tuyến giữa

X

Xích đạo

***Hình 2.12.*** *Phép chiếu và hệ toạ độ lưới UTM*

## **HỆ QUI CHIẾU VN-2000**

## **Hệ tọa độ quy chiếu**

Do đặc điểm về mặt lịch sử, hiện tại bản đồ nước ta được xác lập trên nhiều hệ quy chiếu khác nhau. Trong đó, các bản đồ được dựng trước đây được xây dựng trên hai hệ quy chiếu khác nhau là INDIAN54 và HN72. Hiện nay, hệ quy chiếu VN2000 đã được đưa vào sử dụng thống nhất trên toàn lãnh thổ. Các tham số xác định Ellipsoid quy chiếu và phép chiếu bản đồ tương ứng mỗi hệ quy chiếu như sau:

* ***INDIAN54***
* Ellipsoid quy chiếu:

+ Everest 1830

+ a = 6377276.34518

+ 1/f = 300.801725401854980

* Phép chiếu bản đồ:

+ UTM

+ múi chiếu 60

+ k0 = 0.9996

* ***HN72***
* Ellipsoid quy chiếu:

+ Krasovsky 1940

+ a = 6378245

+ 1/f = 298.300003166221870

* Phép chiếu bản đồ:

+ Gauss-Kruger

+ Múi chiếu 30

+ k0 = 1

* ***VN2000***
* Ellipsoid quy chiếu:

+ WGS 1984

+ a = 6378137

+ 1/f = 298.257222932869640

* Phép chiếu bản đồ:

+ UTM

+ múi chiếu 30

+ k0 = 0.9999

## **Hệ độ cao**

Sau giải phóng miền Nam, năm 1976 Cục Đo đạc – Bản đồ Nhà nước đã tiếp nhận tài liệu mạng lưới độ cao miền Nam lưu giữ tại Nha Địa dư Đà Lạt để đánh giá và đưa vào sử dụng tạm thời. Mạng lưới độ cao hạng I, II miền Nam bao gồm 2.711km đường hạng I và 1.443km đường hạng II, được đo bằng máy thủy chuẩn WILD N3 của Thụy sỹ, BEGAR của Mỹ và Mia Invar. Độ chính xác sau bình sai đạt như sau:

μ = ±21.97mm; m = ±2,2mm.

Độ cao của mạng lưới được tính theo Hệ Mũi Nai (Hà Tiên).

Toàn bộ độ cao đã được chuyển sang Hệ độ cao Hải Phòng –1972 theo công thức hHP = hMN + 0.167m để sử dụng tạm thời.

Geoid là một bề mặt đẳng trọng lực xấp xỉ mực nước biển trung bình. Mô hình geoid được xác lập theo dạng bảng chứa thông tin về giá trị phân cách giữa hai bề mặt geoid và ellipsoid.­ Giá trị phân cách giữa geoid và ellipsoid (N) nhận được trong mô hình geoid là hiệu của cao độ ellipsoid (H) với một điểm đặt trưng riêng. Kết quả của việc dùng mô hình geoid là giá trị cao độ h (elevation) của một điểm phía trên mực nước biển trung bình (mặt geoid).

## **KHÁI NIỆM HỆ THỐNG THÔNG TIN ĐỊA LÝ**

## **Thông tin địa lý**

Thông tin địa lý là những thông tin về các thực thể tồn tại tại một vị trí xác định trên bề mặt Trái Đất ở một thời điểm nào đó. Thông tin địa lý giúp chúng ta nhận biết thuộc tính của một thực thể tồn tại ở đâu và lúc nào. Thí dụ, thông tin về siêu thị, bệnh viện, trường học; thông tin về mạng đường xe buýt; thông tin về một tai nạn giao thông; thông tin về một đám cháy; thông tin về thời tiết; thông tin về những vùng nông nghiệp v.v. . .

Dữ liệu trong các hệ thống thông tin địa lý gồm dữ liệu thuộc tính, dữ liệu không gian và dữ liệu thời gian. Thông tin địa lý rất phong phú, đa dạng và luôn biến động theo thời gian. Tuỳ theo đặc tính của mỗi loại mà tốc độ biến động cũng rất khác nhau. Thông tin địa lý về tự nhiên thường biến đổi chậm hơn thông tin địa lý về kinh tế, xã hội.

## **Hệ thống thông tin địa lý**

Hệ thống thông tin địa lý (Geographic Information System - GIS) là một hệ thống tự động thu thập, lưu trữ, truy vấn, phân tích và hiển thị dữ liệu không gian (Clarke, 1995). Trong đó, phân tích không gianlà khả năng phân biệt GIS với các hệ thống thông tin khác và làm cho GIS có phạm vi ứng dụng trong nhiều lĩnh vực khác nhau.

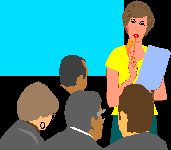
Hệ thống thông tin địa lý là một hệ thống máy tính gồm phần cứng, phần mềm và các thiết bị ngoại vi có khả năng trả lời các câu hỏi cơ bản; “Ai?”, “Cái gì?”, “Ở đâu?”, “Khi nào?”, “Như thế nào?” và “Tại sao?”. Trong đó, câu trả lời cho các câu hỏi: “Ai?”, “Cái gì?” xác định các đối tượng, các hoạt động, các sự kiện cần khảo sát; câu trả lời cho câu hỏi: “Ở đâu xác định vị trí của đối tượng”, các hoạt động hoặc các sự kiện; câu trả lời cho câu hỏi: “Như thế nào/” hoặc “Tại sao?” là kết quả phân tích của hệ thống thông tin địa lý.

Trong tiến trình phát triển kinh tế xã hội, để đạt được một mục đích nào đó, con người cần phải có những quyết định hợp lý và kịp thời. Những quyết định đó thường được thực hiện sau khi thu thập thông tin từ thế giới thực và phân tích xử lý nó theo một quan điểm nào đó. Những quyết định này tác động trực tiếp hoặc gián tiếp trở lại thế giới thực theo khuynh hướng của người quyết định. Nếu quyết định ấy tạo ra nhiều kết quả có lợi hơn cho con người thì được đánh giá là tốt. Ngược lại, nếu quyết định tác động lên thế giới thực sinh ra nhiều hậu quả có hại hơn cho con người thì quyết định ấy bị đánh giá là xấu. Theo quan điểm thông tin, tiến trình này thể hiện một sự tuần hoàn của dữ liệu: Dữ liệu từ thế giới thực được thu thập, lưu trữ, phân tích và ra quyết định. Quyết định tác động trở lại thế giới thực làm thay đổi dữ liệu của thế giới thực. Rồi dữ liệu của thế giới thực lại được thu thập, lưu trữ, phân tích và ra quyết định, vòng tuần hoàn lại tiếp tục. Có như thế, quyết định mới mang lại hiệu quả tốt hơn và hệ thống như vậy mới có ý nghĩa.

Thu thập dữ liệu

*Phân tích và hiển thị dữ liệu*

Ra quyết định



Lưu trữ dữ liệu

Thế giới thực



*Hình 1. Vòng tuần hoàn của dữ liệu địa lý*

Trong một xã hội phát triển, thông tin đặc biệt là thông tin địa lý là một nhu cầu thiết yếu trong mọi lĩnh vực, mọi hoạt động. Ngày nay, GIS đã và đang là công cụ hỗ trợ quyết định hữu hiệu cho mọi hoạt động.

## **Khoa học thông tin địa lý**

Trong tiến trình phát triển, GIS đã được nhiều nhà khoa học sử dụng như công cụ và dành rất nhiều thời gian nghiên cứu, phát triển những công cụ GIS. Thực tế đã hình thành nhóm những nhà khoa học nghiên cứu về GIS. Khoa học thông tin địa lý (Geographic Information Science) ra đời từ những khái niệm do Goodchild (24/02/1944) đưa ra từ 1992 nhằm thúc đẩy và định hướng các hoạt động nghiên cứu về GIS với các định nghĩa sau:

* + Khoa học thông tin địa lý là một khoa học sử dụng các hệ thống thông tin địa lý như những công cụ để hiểu biết về thế giới thực bằng cách mô tả và giải thích mối quan hệ của con người với thế giới thực.
  + Khoa học thông tin địa lý là một khoa học sử dụng và nghiên cứu các phương pháp, các công cụ để thu thập, lưu trữ, phân phối, phân tích, hiển thị và khám phá thông tin không gian.

Khoa học thông tin địa lý hướng tới những bài toán phân tích, mô hình hóa và dự báo không gian được phát triển trên cơ sở sự phát triển của khoa học máy tính. Khoa học thông tin địa lý sử dụng và phát triển các mô hình toán để biểu diễn, lưu trữ, phân tích, hiển thị dữ liệu về các đối tượng, các sự kiện, các hiện tượng trong thế giới thực. Để phân tích về mối quan hệ giữa con người với thế giới thực, khoa học thông tin địa lý cũng sử dụng và phát triển các mô hình toán máy tính để xử lý dữ liệu không gian, phân tích ý nghĩa và quan hệ của các thực thể không gian theo các lý luận của khoa học địa lý. Khoa học thông tin địa lý là một khoa học liên ngành của khoa học máy tính, khoa học toán và khoa học địa lý.

**Khoa học toán**

**Khoa học địa lý**

**Khoa học máy tính**

**GIScience**

1. Hình 2. GIS là một khoa học liên ngành (Nguồn [1])

Từ khi ra đời, khoa học thông tin địa lý phát triển không ngừng cùng với sự phát triển của khoa học máy tính và công nghệ thông tin. Với sự phát triển đó, công nghệ thông tin địa lý ngày càng được nhiều ngành nghề, nhiều lĩnh vực sử dụng như một công cụ để giải quyết những bài toán đặc thù của từng chuyên ngành. Sự ứng dụng rộng rãi đó đã làm tăng thêm sức sống cho khoa học thông tin địa lý, vì khi sử dụng công nghệ GIS vào các chuyên ngành khác nhau đó, khoa học GIS nhận thêm được nhiều bài toán cần phải nghiên cứu.

## **TIẾN TRÌNH PHÁT TRIỂN GIS**

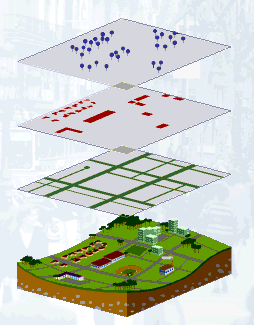
Hệ thống thông tin địa lý ra đời từ đầu thập niên 60 trong các cơ quan địa chính ở Canada. Trong suốt thời gian hai thập niên 60 và 70, GIS chỉ được một vài cơ quan chính quyền khu vực Bắc Mỹ quan tâm nghiên cứu. Mãi đến đầu thập niên 80, khi phần cứng máy tính phát triển mạnh với hiệu năng cao, giá thành rẻ; đồng thời sự phát triển nhanh về lý thuyết và ứng dụng cơ sở dữ liệu, thuật toán nhận dạng, xử lý ảnh cùng với nhu cầu cần thiết về thông tin đã làm cho công nghệ GIS càng được quan tâm.

Cuối thế kỷ 20, trên thế giới đã hình thành nhiều cơ quan nghiên cứu GIS với quy mô lớn, nhiều hướng tiếp cận và mục tiêu khác nhau:

* RRL (Regional Research Laboratory) được thành lập vào tháng 2/1980 ở Anh, là trung tâm nghiên cứu về quản lý cơ sở dữ liệu, phát triển phần mềm và phân tích không gian.
* NCGIA (National Central for Geographic Information and Analysis) thành lập từ năm 1988 tại Hoa Kỳ, tập trung vào 5 hướng nghiên cứu: Phân tích và thống kê không gian; quan hệ giữa không gian và cấu trúc dữ liệu; trí tuệ nhân tạo và các hệ chuyên gia; trình bày hình ảnh; những đề tài kinh tế, xã hội, văn hoá. Những đề tài của NCGIA mang nặng tính hàn lâm, đi sâu vào bản chất kỹ thuật và phương pháp luận.
* NEXPRI (Dutch Expertise Central for Spatial Data Analysis) thành lập vào tháng 1/1989 ở Hà Lan, nhắm vào 4 hướng nghiên cứu chính là: lý thuyết về phân tích không gian; đánh giá định lượng về đất; sự di chuyển của vật chất và ô nhiễm; phát triển những phương pháp và kỹ thuật GIS.
* GISDevelopment là một tổ chức phi chính phủ của những chuyên gia GIS châu Á, hoạt động như một mạng lưới thông tin về sự phát triển khoa học, công nghệ và ứng dụng công nghệ thông tin địa lý tại các nước châu Á. GISDevelopment có tạp chí GISDevelopment xuất bản hàng tháng, website www.GISDevelopment.net và tổ chức hội nghị MapAsia mỗi năm một lần với sự tham gia của nhiều nhà khoa học GIS đến từ các nước châu Á.
* Ở Việt Nam, trung tâm Công nghệ Thông tin Địa lý – DITAGIS – được thành lập từ tháng 10/1994, là trung tâm chuyên nghiên cứu và phát triển ứng dụng khoa học và công nghệ thông tin địa lý (GIScience and GISystems). Năm 1995, Bộ Khoa học và Công nghệ đã liên tục đầu tư nhiều dự án và đề tài nghiên cứu GIS, từ những đề tài nghiên cứu phương pháp luận đến những đề tài nghiên cứu ứng dụng.

## **CHỨC NĂNG CỦA GIS**

Hệ thống thông tin địa lý có bốn chức năng cơ bản là thu thập, lưu trữ, phân tích và hiển thị dữ liệu.



*Nhập*

*Lưu trữ*

*Phân tích*

*Hiển thị*

*Hình 3.Các chức năng của hệ thống thông tin địa lý*

## **Thu thập dữ liệu**

Dữ liệu địa lý là thành phần quan trọng và tồn tại lâu bền trong trong hệ thống thông tin địa lý. Hầu hết các phần mềm GIS đều cung cấp chức năng để nhập dữ liệu vào hệ thống.

Dữ liệu của một hệ thống thông tin địa lý có thể được cung cấp từ bản đồ giấy, số liệu ghi nhận trên giấy, ảnh vệ tinh hoặc ảnh được chụp từ máy bay, các thiết bị đo đạc kỹ thuật số, các thiết bị định vị mặt đất, các thiết bị định vị vệ tinh (GPS: Global Position System), hệ thống thu thập dữ liệu tự động (SCADA: Supervisory Control And Data Acquisition),…

## **Lưu trữ dữ liệu**

Các đối tượng không gian địa lý có thể được biểu diễn theo mô hình vector hoặc raster.



x,y

x,y

x,y

x,y



***Cấu trúc Vector***



***Cấu trúc Raster***

x,y

*Pixel*

*Cột*

*Dòng*

Hình 4. Biểu diễn dữ liệu trong cấu trúc Vector và Raster

Mô hình vector: Mô hình dữ liệu vector biểu diễn các đối tượng địa lý trên mặt đất bằng những điểm, đường, vùng trong mặt phẳng tọa độ Descartes. Mỗi điểm được xác định bởi cặp toạ độ (x, y), mỗi đường được tuyến tính hóa từng đoạn, biểu diễn bằng một chuỗi những cặp toạ độ (xi, yi), một vùng được xác định bởi một đường khép kín và được biểu diễn bằng một chuỗi cặp toại độ (xi, yi) có toạ độ điểm đầu và toạ độ điểm cuối trùng nhau.

* Mô hình raster: Mô hình dữ liệu ấn định vị trí của các đối tượng không gian vào các ô lưới hình ô vuông (hoặc chữ nhật hay tam giác nhưng rất ít được sử dụng) có kích thước bằng nhau gọi là pixel, được xác định bằng toạ độ (x, y) là số thứ tự của hàng, cột của pixel. Trong cấu trúc raster, đường được biểu diễn bằng những pixel có cùng giá trị thuộc tính f(x,y) kề nhau, trải rộng ra theo một phương nào đó. Vì trong cấu trúc raster, các pixel được xếp theo hàng, cột như một ma trận điểm nên đường ở đây không trơn, có dạng zic-zac. Vùng được xác định bằng một mảng gồm nhiều pixel có cùng giá trị thuộc tính f(x,y) kề nhau, trải rộng ra theo nhiều phương.

Dữ liệu thuộc tính có thể được lưu trữ gắn kết trong mỗi bảng thuộc tính của đối tượng không gian hoặc là các bảng dữ liệu hoàn toàn độc lập, khi cần thiết thì bảng dữ liệu này được kết nối vào bảng thuộc tính của đối tượng không gian tạo thành dữ liệu địa lý.

Dữ liệu của một hệ thống thông tin địa lý được tổ chức lưu trữ như những tệp dữ liệu (data file) cho một người dùng hoặc cơ sở dữ liệu (personal database), cơ sở dữ liệu địa lý (geodatabase) cho nhiều người dùng (multiusers).

Dữ liệu lưu trữ trong một hệ thống thông tin địa lý được truy vấn theo hai phương thức:

* Truy vấn từ đối tượng không gian để tìm thuộc tính. Trong cách truy vấn này, người sử dụng phải xác định được vị trí của đối tượng cần quan tâm, sau đó xem xét thuộc tính của chúng.
* Truy vấn từ dữ liệu thuộc tính để tìm vị trí của đối tượng trong không gian bằng cách xây dựng các biểu thức dựa vào các điều kiện ràng buộc. Trong trường hợp này, người sử dụng đã biết các đặc điểm của đối tượng và muốn tìm ra vị trí của các đối tượng đó.

## **Phân tích dữ liệu**

Cùng với khả năng của máy tính và toán học, hệ thống thông tin địa lý đã cung cấp nhiều phương tiện để thực hiện những bài toán phân tích không gian. Những thuật toán phân tích trên một lớp dữ liệu, chồng xếp nhiều lớp dữ liệu, phân tích mạng,… là những thuật toán hỗ trợ tích cực trong các bài toán quản lý, quy hoạch, kế hoạch của nhiều lĩnh vực tài nguyên, môi trường, đất đai, cơ sở hạ tầng, kinh tế, y tế, giáo dục, thương mại, dịch vụ,…

Phân tích dữ liệu là khả năng trả lời những câu hỏi về sự tác động lẫn nhau của những mối quan hệ không gian và thuộc tính giữa nhiều tập dữ liệu. Có nhiều phương pháp phân tích dữ liệu trong GIS, tuỳ vào từng mục tiêu và nguồn dữ liệu cụ thể mà ta có thể chọn các phương pháp phân tích khác nhau:

* Thao tác phân tích trên một lớp dữ liệu (Single Layer Operations) là những thuật toán xử lý dữ liệu trên một lớp như thuật toán buffer, truy vấn thuộc tính từ thuộc tính, truy vấn thuộc tính từ không gian, truy vấn không gian từ thuộc tính hoặc tạo những tập dữ liệu mới được thực hiện trên một lớp dữ liệu.
* Thao tác phân tích trên nhiều lớp dữ liệu (Multiple Layer Operations) là những thao tác trên nhiều lớp dữ liệu không gian để thực hiện các thuật toán phân tích: chồng lớp (union, intersect, identify), phân tích gần kề, phân tích tương quan không gian.
* Mô hình hoá không gian (Spatial Modeling) là xây dựng những mô hình để giải thích và dự báo theo không gian, mô phỏng không gian, thuật toán nội suy không gian.
* Phân tích mẫu điểm (Point Pattern Analysis) thực hiện các thuật toán phân tích số đông trên những lớp dữ liệu không gian điểm.
* Phân tích mạng (Network Analysis) ứng dụng vào những đối tượng dạng đường, những đối tượng này được tổ chức trong mạng lưới liên kết.
* Phân tích bề mặt (Surface Analysis) bao gồm những thuật toán phân tích, cần đến thứ nguyên thứ ba của những biến phân bố trong không gian.

## **Hiển thị dữ liệu**

Chức năng hiển thị trong hệ thống thông tin địa lý là biến ngôn ngữ của máy tính thành ngôn ngữ thân thiện với người dùng, hiển thị những kết quả truy vấn hoặc phân tích không gian. Dữ liệu GIS được thị lên màn hình máy tính hoặc bằng văn bản để cung cấp thông tin cho người dùng. Trong GIS người ta dùng hình ảnh, hình vẽ, mô hình trực quan, chữ viết, biểu đồ, bản đồ, bảng thống kê, ký hiệu, màu sắc và cả âm thanh để trình bày vị trí và thuộc tính của các đối tượng và các kết quả phân tích.

Ngôn ngữ và biểu tượng sử dụng trong bản đồ đã được xây dựng, phát triển và sử dụng từ nhiều năm nay. Do đó, khi hiển thị cho người dùng ngoài hệ thống, những người dùng trong hệ thống cần phải biên tập sao cho ngôn ngữ trên bản đồ phải gần gũi với ngôn ngữ tự nhiên, biểu tượng quen thuộc và dễ hiểu đối với mọi người.

## **CÁC THÀNH PHẦN CỦA HỆ THỐNG THÔNG TIN ĐỊA LÝ**

GIS có bốn thành phần cơ bản: dữ liệu, phần cứng, phần mềm, con người.

## **Thành phần dữ liệu**

Dữ liệu GIS bao gồm dữ liệu không gian và dữ liệu phi không gian còn gọi là dữ liệu thuộc tính hay dữ liệu hình học, được thu thập, lưu trữ theo một cấu trúc chuẩn, có thể liên thông trên mạng và được bảo quản theo một chế độ nhất định. Tại đây, có những bài toán đặt ra như mô hình cơ sở dữ liệu phục vụ các yêu cầu lưu trữ, truy xuất, hiển thị, công nghệ nhập dữ liệu, phân quyền sử dụng dữ liệu, an ninh dữ liệu trên mạng, tổ chức cơ sở dữ liệu phân tán hay tập trung, v.v... Cơ sở dữ liệu là một thành phần quan trọng trong các hệ thống thông tin địa lý phục vụ quản lý hành chính nhà nước. Dữ liệu không gian là biểu diễn hình học của các đối tượng địa lý liên kết với vị trí trên thế giới thực. Những đối tượng địa lý được biểu diễn bởi điểm, đường và vùng. Dữ liệu thuộc tính là những thuộc tính mô tả đặc điểm của các đối tượng địa lý. Trong hệ thống thông tin địa lý, dung lượng dữ liệu thường rất lớn, cần được cập nhật thường xuyên và phải được tổ chức lưu trữ theo một mô hình cơ sở dữ liệu được thiết kế để phục vụ cho công tác lưu trữ, quản lý, truy vấn và hiển thị dữ liệu. Ở đây, giá trị của dữ liệu cũng cần được xác lập. Cụ thể, dữ liệu trong hệ thống thông tin địa lý phục vụ quản lý hành chính nhà nước là một loại tài nguyên quốc gia, trong đó có những loại dữ liệu phổ biến, những loại dữ liệu chia sẻ có điều kiện và những loại dữ liệu không thể chia sẻ theo qui định bảo mật của nhà nước.

Dữ liệu trong một hệ thống thông tin địa lý bao gồm dữ liệu không gian, dữ liệu phi không gian liên kết nhau có thể được quản lý qua mạng theo mô hình tập trung hay phân tán. Cơ sở dữ liệu của hệ thống thông tin địa lý bao gồm hai phần chính là cơ sở dữ liệu nền và cơ sở dữ liệu chuyên ngành.

a. Cơ sở dữ liệu nền

Cơ sở dữ liệu nềnbao gồm những lớp dữ liệu mà hệ thống thông tin địa lý chuyên ngành nào cũng sử dụng như dữ liệu về lưới tọa độ (tọa độ địa lý, tọa độ quốc gia), dữ liệu về giao thông, dữ liệu về thủy văn, dữ liệu về độ cao, dữ liệu về hành chính, dữ liệu về giải thửa, v.v...

b. Cơ sở dữ liệu chuyên ngành

Cơ sở dữ liệu chuyên ngành bao gồm dữ liệu của các thực thể chuyên ngành được biểu diễn theo mô hình dữ liệu không gian và phi không gian liên kết với nhau. Có thể có cơ sở dữ liệu của các chuyên ngành như tài nguyên thiên nhiên, môi trường, sử dụng đất, cơ sở hạ tầng kỹ thuật, các chuyên ngành kinh tế xã hội. Khi thiết kế cơ sở dữ liệu chuyên ngành cần chú ý đến những quan hệ giữa các thực thể trong một chuyên ngành đồng thời những quan hệ của các thực thể giữa các ngành với nhau. Đối với mỗi chuyên ngành, tùy theo mục tiêu của hệ thống thông tin địa lý, những qui trình xây dựng cơ sở dữ liệu được thiết kế khác nhau.

Dữ liệu GIS thường được tổ chức theo từng lớp và theo từng chủ đề riêng biệt. Hầu hết các phần mềm GIS đều có chức năng lưu trữ dữ liệu theo từng lớp và theo loại đối tượng; điểm, đường, vùng, tuỳ theo phần mềm mà có tên gọi khác nhau cho các lớp chủ đề khác nhau: trong phần mềm Arcview gọi là Theme, Arc/Info gọi là Coverage, MapInfor gọi là Table, ArcGIS gọi là Layer.

## **Phần cứng**

Phần cứng bao gồm hệ thống máy tính và các thiết bị thu thập, nhập, xuất, lưu trữ dữ liệu. Trong đó, hệ thống máy tính gồm chỉ có một hoặc nhiều máy tính được kết nối thành một hệ thống mạng; các thiết bị ngoại vi để thu thập, nhập, xuất dữ liệu như máy digitizer, scanner, máy in, plotter, máy ghi đĩa quang hoặc CD, băng từ, và các thiết bị định vị khác, v.v...

Tùy theo mục tiêu và qui mô tổ chức hệ thống thông tin địa lý, nhà thiết kế sẽ xác định qui mô và cấu hình phần cứng thích hợp. Phần cứng được thiết kế như một hệ thống mạng máy tính cục bộ (LAN) hoặc mạng diện rộng (WAN), tùy theo nhiệm vụ của hệ thống. Các thiết bị cập nhật dữ liệu thường xuyên, các thiết bị lưu trữ, hiển thị cũng được trang bị đáp ứng mục tiêu, nhiệm vụ của hệ thống.

Trong tiến trình xây dựng hệ thống, ta cần phân chia các giai đoạn và trang bị phần cứng hệ thống cho phù hợp với nhu cầu khai thác vì giá thành của các thiết bị càng ngày càng giảm, trong khi tính năng kỹ thuật của thiết bị tăng lên. Đặc biệt lưu ý rằng những thiết bị nhập dữ liệu bản đồ giấy chỉ sử dụng trong một thời gian ngắn trong thời gian chuyển tiếp từ công nghệ cũ sang công nghệ thông tin địa lý. Ngày nay, những thiết bị thu thập dữ liệu không gian có độ chính xác và khả năng tự động cao đang dần thay thế những thiết bị đo đạc truyền thống. Do đó, các hệ thống GIS cũng cần thường xuyên đổi mới và thay thế trang bị thiết bị phục vụ cho nhu cầu cập nhật dữ liệu.

Đối với một tỉnh/thành phố, nhu cầu số hóa bản đồ không phải là nhu cầu lớn, số hóa bản đồ chẳng những không phải là mục tiêu xây dựng hệ thống thông tin địa lý mà đó chỉ là một phần trong công đoạn khởi tạo cơ sở dữ liệu nền cho hệ thống thông tin địa lý. Công nghệ số hóa bản đồ chỉ là một công nghệ quá độ từ công nghệ truyền thống, sử dụng và lưu trữ dữ liệu không gian trên bản đồ giấy, chuyển sang công nghệ thông tin địa lý với những thiết bị kỹ thuật số có khả năng thu thập dữ liệu không gian với độ chính xác cao. Những sự đầu tư thiết bị số hóa bản đồ trong thời gian này sẽ nhanh chóng lạc hậu.

## **Phần mềm**

Phần mềm GIS bảo đảm đủ 4 chức năng của hệ thống thông tin địa lý là nhập, lưu trữ, phân tích và xử lý, hiển thị dữ liệu không gian, phi không gian. Ngoài ra, phần mềm còn phải đáp ứng yêu cầu của hệ thống mở, cho phép nâng cấp khi cần thiết hoặc/và có thể liên kết với các hệ thống khác, đặc biệt là hệ thống cơ sở dữ liệu thuộc tính hiện có. Hiện nay, trên thế giới có rất nhiều phần mềm khác nhau, mỗi một phần mềm đều có điểm mạnh riêng, tuỳ theo mục tiêu khai thác mà người sử dụng có thể quyết định trang bị phần mềm hợp lý. Phần mềm của hệ thống cần được trang bị từng bước và phù hợp với chức năng, nhiệm vụ của từng vị trí trong hệ thống, không nhất thiết phải trang bị những phần mềm quá lớn hoặc quá nhỏ. Ở đây cũng cần quan tâm đến khả năng mở rộng, tương thích của phần mềm và định dạng dữ liệu do phần mềm tạo ra. Trong những trường hợp cần thiết, cũng có thể thiết kế bổ sung những phần mềm chuyển đổi định dạng dữ liệu.

Một hệ thống GIS bao gồm nhiều môđun phần mềm, có các thành phần chính sau:

* Công cụ nhập và thao tác trên các thông tin địa lý
* Hệ quản trị CSDL (DBMS)
* Công cụ hỗ trợ hỏi đáp, phân tích và hiển thị địa lý
* Giao diện đồ hoạ người-máy (GUI) để truy cập các công cụ dễ dàng.

Trong đó khả năng lưu trữ, quản lý dữ liệu không gian bằng *hệ quản trị CSDL địa lý* là khía cạnh quan trọng nhất của GIS. Các môđun khác là công cụ phân tích dữ liệu, xây dựng báo cáo và truyền tin.

## **Con người**

Con người chia thành 3 nhóm chính: Chuyên gia về GIS làm nhiệm vụ nghiên cứu các công cụ, xây dựng phần mềm GIS, thiết kế hệ thống GIS, v.v... Nhóm thứ hai là người thu thập, xử lý, biên tập, quản lý và bảo dưỡng dữ liệu GIS, chịu trách nhiệm chính về sự chính xác, tính cập nhật của dữ liệu. Nhóm cuối cùng là những người khai thác thông tin phục vụ cho các mục tiêu nghiên cứu của mình.

Cả bốn thành phần trên có mối quan hệ mật thiết với nhau, tương tác với nhau để GIS hoàn thành và phát triển. Trên hình 1.4 mô tả các mối quan hệ của 4 thành phần GIS. Sự hiểu biết một cách đầy đủ các thành phần của GIS rất quan trọng, bởi vì nếu thiếu hoặc một thành phần của chúng phát triển không tương xứng với các thành phần khác sẽ làm cho GIS hoạt động không có hiệu quả, thậm chí sẽ “chết”.

**Phần cứng:**

* Máy tính
* Thiết bị ngoại vi

**Phần mềm:**

* Hệ thống
* Ứng dụng
* Hệ quản lý CSDL

GIS

**Dữ liệu:**

* Hình học
* Thuộc tính

**Con người:**

* Chuyên gia GIS
* Quản lý hệ thống
* Khai thác

Hình 5. Các thành phần của GIS

## **CÁC LĨNH VỰC ỨNG DỤNG GIS**

Nhờ khả năng phân tích và xử lý đa dạng, kỹ thuật GIS hiện nay được ứng dụng phổ biến trong nhiều lĩnh vực, được xem là “công cụ hỗ trợ ra quyết định. Một số lĩnh vực được ứng dụng chủ yếu ở nhiều nơi trên thế giới như địa chất, nông nghiệp, đô thị, giao thông, ngân hàng, thực vật, địa chính, kinh tế, môi trường, v.v...

Hiện nay đã có phần mềm ứng dụng GIS vào các chuyên đề: cấp thoát nước, công nghiệp, cung cấp điện, dân cư, giao thông, giáo dục, môi trường, nhà đất, quy hoạch, thương mại - dịch vụ, tôn giáo, văn hóa, xây dựng, y tế…

## **KHUYNH HƯỚNG PHÁT TRIỂN GIS**

### **Khuynh hướng phát triển về lý thuyết**

Hiện nay, nhiều nhà khoa học đang nghiên cứu phương pháp biểu diễn dữ liệu không gian trong các hệ thống thông tin địa lý, sự liên quan giữa các loại dữ liệu bao gồm dữ liệu không gian, dữ liệu thuộc tính và dữ liệu thời gian. Mối quan hệ các bài toán phân tích không gian theo thời gian thực. Phân tích thống kê dữ liệu không gian. Thiết kế mô hình dữ liệu và cấu trúc dữ liệu thích hợp. Nghiên cứu phương pháp và kỹ thuật thiết kế cơ sở dữ liệu không gian. Đào tạo và huấn luyện kỹ thuật sử dụng GIS. Nghiên cứu đầy đủ hơn về công nghệ bản đồ, truyền thông bản đồ.

### **Khuynh hướng phát triển phần mềm**

* Hệ quản trị CSDL
* GIS với kỹ thuật đa phương tiện (multimedia): Kỹ thuật đa phương tiện tích hợp biểu diễn thông tin với các dạng dữ liệu khác nhau như hình ảnh, âm thanh, chữ viết rất phù hợp để trình diễn các hệ thống thông tin địa lý phục vụ giảng dạy các môn học có liên quan đến thông tin không gian, đánh giá tác động môi trường, theo dõi biến động môi trường, mô phỏng các tiến trình địa lý - địa chất…
* GIS thông minh: Những định hướng mới nhất trong việc nghiên cứu các quá trình xử lý thông tin địa lý cho thấy nhu cầu cần tìm ra cách tốt hơn để mô tả tính bất thường của thế giới thực và những phương pháp mới để có thể suy đoán chính xác. Áp dụng các kỹ thuật thông minh nhân tạo như logic mờ, hệ chuyên gia, mạng nơ-ron cũng có thể được phát triển trong GIS để tăng khả năng giao tiếp với người sử dụng.
* Mô phỏng và trợ giúp quyết định: Xác định những hệ thống thông tin địa lý thời gian thực là mục tiêu quan trọng mà tất cả những nhà khoa học, kỹ thuật của nhiều ngành có liên quan như kỹ thuật phần cứng, toán và kỹ thuật phần mềm đang theo đuổi và hỗ trợ cho nhau. Đặc biệt, dự báo trong nhiều lĩnh vực khác nhau là bài toán được quan tâm trong các hệ thống thông tin địa lý trong thời gian tới.

### **Khuynh hướng phát triển ứng dụng**

Con người với mong muốn được hiểu biết về các thực thể, các sự kiện, hiện tượng như thế nào, xảy ra ở đâu, khi nào và tại sao. Nhà nghiên cứu địa chất muốn biết tính chất về địa chất, nhà môi trường muốn biết thông tin về tình trạng ô nhiễm đất, nước và không khí trên thực thể đó. Các nhà quy hoạch cần đưa ra những quyết định vùng nào sử dụng cho mục đích gì là tốt nhất. Còn những nhà kinh doanh cần có thông tin địa lý để qui hoạch chiến lược thị trường, tiêu thụ hàng hóa được nhanh chóng…

## **BÀI TẬP CHƯƠNG 1**

1. Trình bày khái niệm GIS?
2. Trình bày các thành phần của GIS?
3. GIS có những chức năng gì?
4. Phân tích chức năng nhập dữ liệu ?
5. Phân tích chức năng lưu trữ và quản lý dữ liệu?
6. Phân tích chức năng phân tích dữ liệu?
7. Phân tích chức năng hiển thị và xuất dữ liệu?
8. Trình bày chức năng phân tích không gian GIS? Lấy ví dụ minh họa?
9. Trình bày các lĩnh vực ứng dụng của GIS ? Lấy ví dụ minh họa ?
10. Nêu ứng dụng của GIS trong đời sống?

# 

# ***CHƯƠNG 2*** MÔ HÌNH DỮ LIỆU CỦA HỆ THỐNG THÔNG TIN ĐỊA LÝ

Dữ liệu về các đối tượng không gian có thể được xây dựng theo mô hình Raster hoặc vector với các cấu trúc tương ứng. Cấu trúc raster được sử dụng rộng rãi trong các hệ thống xử lý ảnh và GIS raster. Các cấu trúc Vector chiếm ưu thế trong các hệ thống CAD và GIS với những tiềm năng vẽ bản đồ mạnh mẽ. Nhiều cấu trúc khác nhau được phát triển cho cả hai mô hình Vector và Raster.

Việc lựa chọn một cấu trúc dữ liệu tốt nhất tùy thuộc vào bản chất dữ liệu và mục đích sử dụng. Các cấu trúc khác nhau được sử dụng theo những tác vụ khác nhau tùy thuộc độ phù hợp và tính hiệu quả của việc sử dụng. Ví dụ, một tập điểm mẫu với các thuộc tính liên quan có thể tồn tại ở một trong những dạng sau:

1. Dạng bảng thuộc thuộc tính với các tọa độ không gian.
2. Dạng raster chứa các giá trị được nội suy từ các điểm đối với một trong số các thuộc tính.
3. Dạng vector, biểu diễn các biên giới của những vùng trên một bản đồ đường bao được tạo ra từ Raster.
4. Dạng cấu trúc TIN.

Mỗi cấu trúc đều có ưu điểm, nhược điểm riêng và khả năng chuyển từ một cấu trúc này sang cấu trúc kia là rất quan trọng.

## **CẤU TRÚC DỮ LIỆU RASTER**

Mô hình raster chia không gian thành những ô lưới hình vuông (hoặc chữ nhật, cũng có thể hình tam giác nhưng rất ít được sử dụng) có kích thước bằng nhau gọi là điểm ảnh (pixel) được xác định vị trí bằng toạ độ (x,y) là số thứ tự của hàng và cột của pixel. Điểm ảnh là phần tử cơ sở của cấu trúc dữ liệu raster để biểu diễn một đặc trưng địa lý f(x,y) nào đó, nghĩa là dữ liệu trong mỗi điểm ảnh là đồng nhất. Liên hệ với thế giới thực, kích thước mỗi ô sẽ tương đương với một kích thước nào đó trong thế giới thực, thí dụ điểm ảnh (1m x 1m), (100m x 100m), v.v...

Trong cấu trúc raster, đường được biểu diễn bằng những pixel có cùng giá trị f(x,y) liên tiếp nhau. Vì trong cấu trúc raster, các pixel được xếp theo hàng, cột như một ma trận điểm nên đường ở đây không trơn mà có dạng zic-zac. Vùng được xác định bằng một mảng gồm nhiều điểm ảnh có cùng giá trị thuộc tính f(x,y) trải rộng ra theo nhiều phương.

Dữ liệu không gian khi lưu trữ theo mô hình raster phụ thuộc vào các yếu tố sau:

* Độ phân giải không gian: xác định kích thước nhỏ nhất của một lớp dữ liệu (ảnh) trong hệ thống. Qua đó thể hiện được không gian nhỏ nhất mà một ảnh trong hệ thống có thể quan sát được.
* Độ phân giải pixel: xác định kích thước nhỏ nhất của một pixel và thường được tính toán dựa trên số lượng pixel trên một đơn vị chiều dài.
* Độ phân giải mã hoá: khoảng cách nhỏ nhất tính theo đơn vị mức xám giữa hai giá trị mức xám liền kề (fi+1 – f i ).

C

D

D

C

C

C

A

D

D

C

D

D

C

N

N

N

D

D

D

D

D

Cấu trúc dữ liệu raster có hai đặc điểm cần lưu ý là:

* Mỗi điểm ảnh chỉ biểu diễn một thuộc tính, xác định bởi giá trị f(x,y). Do đó, mỗi lớp dữ liệu chỉ biểu diễn một thuộc tính, lớp f. Ví dụ, các lớp dữ liệu thổ nhưỡng và thực vật phải được xây dựng riêng rẻ với các giá trị thuộc tính lần lượt là f(x,y) và g(x,y). Khi tích hợp thông tin của hai hoặc nhiều lớp dữ liệu, lớp dữ liệu tích hợp được thành lập với giá trị h(x,y) của điểm ảnh được xác định bằng cách tích hợp các giá trị f(x,y) và g(x,y) của các pixel có cùng vị trí của các lớp dữ liệu thành phần theo một phép tính thích hợp.
* Khi thay đổi độ phân giải (kích thước điểm ảnh thay đổi), dung lượng dữ liệu thay đổi theo. Dung lượng dữ liệu tăng theo bình phương tỉ lệ gia tăng độ phân giải (độ phân giải tăng đồng nghĩa với kích thước điểm ảnh giảm).

### **Dạng hình học và topology của dữ liệu Raster**

*a. Đối tượng điểm (Point objects)*

Trong cấu trúc raster, mỗi đối tượng điểm là một điểm ảnh được gán ID điểm, vị trí của điểm là toạ độ (i,j) của điểm ảnh.

*Hình 19. Dạng hình học và Topology của đối tượng điểm*

**Số pixel i**

**Số hàng j**

**(i,j) = (5,3);(7,5);(8,2)**

*b. Đối tượng đường (Line objects)*

Trong cấu trúc raster, mỗi đối tượng đường được gán ID đường, hình dạng là tập hợp các điểm ảnh tạo ra đường và vị trí được xác định bởi tọa độ của các điểm ảnh.

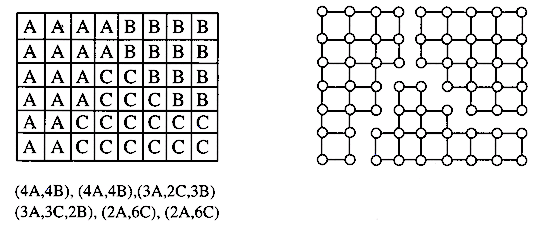
*Hình 20. Hình học và Topology đối tượng đường*

(1,3);(2,2);(3,2) ;(4,3); (5,4) ;(6,5) ;(7,5) ;(8,4)

### 

*c. Đối tượng vùng (Polygon objects)*

Trong cấu trúc raster, mỗi đối tượng vùng được gán ID vùng, hình dạng của vùng được xác định bởi tập các điểm ảnh thuộc vùng đó và vị trí được xác định bởi toạ độ của các điểm ảnh.



## 

### **Các thuật toán nén dữ liệu cấu trúc raster**

Để giảm dung lượng khi lưu trữ, dữ liệu ảnh raster có thể được nén theo các giải thuật mã hóa khác nhau như mã run-length, cây tứ phân.

*a. Mã Run –length*

Mã run - length lợi dụng sự giống nhau của các pixel lân cận để biểu diễn chúng từng loạt pixel có giá trị giống nhau theo từng hàng từ trái qua phải, từ trên xuống dưới được biểu diễn bằng một cặp số (chiều dài chuỗi, giá trị của pixel).

Mã này rất hữu hiệu khi thuộc tính của lớp dữ liệu ảnh (giá trị của pixel) ít thay đổi. Ngược lại, nếu giá trị thuộc tính thay đổi nhiều, sự hữu hiệu của mã không cao mà đôi khi còn làm tăng dung lượng dữ liệu.Mã run-length tuy có thuật toán nén và giải mã đơn giản nhưng tỉ số nén không cao, thao tác tìm kiếm thường chậm vì lưu trữ theo hàng, không lưu ý tới tính lân cận. Để khắc phục tình trạng này, mã run-length được cải biên theo những đường đọc dữ liệu khác nhau:

Biểu diễn mã run-length của lớp dữ liệu ảnh raster hình bên: (10,A) (10,A) (4,A) (6,B) (3,A) (7,B) (4,D) (6,B) (5,D) (5,B) (5,D) (5,C) (5,D) (5,C) (5,D) (5,C) (5,D) (5,C).

*Hình 24. Mã Run-length*

B

1

2

3

4

0

A

B

B

A

A

A

A

2

A

A

A

A

1

A

6

A

A

3

B

B

BC

A

A

5

A

5

4

A

A

A

A

88

7

0

B

B

A

A

A

A

8

7

6

B

98

98

A8

AA8

B8

A

A8

B

B

B

B

B

B

B

B

B

B

B

B

B

B

D

D

D

D

D

D

D

D

D

D

D

D

D

D

D

D

D

D

D

D

D

D

D

D

CD

CD

CD

CD

CD

CD

CD

CD

CD

CD

CD

CD

CD

CD

CD

CD

CD

CD

CD

CD

D

D

D

D

D

Hướng đọc các pixel theo mã run-length

(24,A) (13,B) (3,A) (13,B) (4,D) (11,B) (10,D) (10,C) (10,D) (10,C) (5,D).

*Hình 25. Mã Run-length cải biên*

### 

*b. Mã cây tứ phân*

3

0

0

2

1

2

0

1

3

1

Mã tứ phân cũng dựa theo nguyên lý sự giống nhau của nhiều pixel trong những vùng có cùng thuộc tính. Mỗi vùng chỉ cần gán một giá trị. Vùng được xác định bằng cách chia nhỏ diện tích làm 4 phần bằng nhau, mỗi phần lại được chia nhỏ thành 4 phần, v.v.... Việc chia nhỏ này sẽ không thực hiện ở những vùng có cùng thuộc tính.

## **CẤU TRÚC DỮ LIỆU VECTOR**

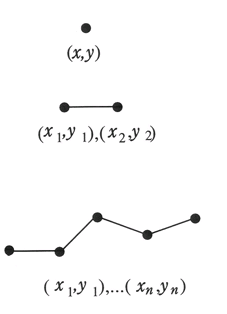
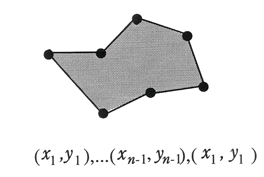
Các đối tượng không gian khi biểu diễn ở cấu trúc vector thường tổ chức dưới dạng điểm, đường và vùng trên một hệ thống tọa độ xác định. Mỗi điểm được xác định bởi một cặp toạ độ (x,y); đường được xác định bởi một chuỗi liên tiếp các điểm và vùng được xác định bởi các đường khép kín.

Trong cấu trúc dữ liệu vector có hai loại cấu trúc được biết đến đó là cấu trúc Spaghetti và cấu trúc Topology, cấu trúc Spaghetti ra đời trước và hiện nay vẫn được sử dụng nhiều trong các phần mềm GIS.

### **Tính hình học của cấu trúc vector**

Các đối tượng trong không gian được phân loại thành 3 dạng

* Dạng đối tượng điểm (point): Điểm dùng cho tất cả các đối tượng không gian được biểu diễn như một cặp tọa độ (x,y). Ví dụ như điểm cao độ, điểm nhà máy, điểm quan trắc, …
* Dạng đối tượng đường (line/polyline) dùng để biểu diễn tất cả các đối tượng có dạng tuyến, được tạo nên từ hai hoặc nhiều cặp tọa độ (x,y). Ví dụ: đường giao thông, đường bình độ, đường ống, đường dây điện, dây điện thoại, …
* Dạng đối tượng vùng (region): Vùng là một đối tượng hình học hai chiều



Các đối tượng trong không gian được trình bày theo dạng hình học dưới hình thức điểm, đường và vùng một cách riêng biệt. Cấu trúc hình học của điểm là toạ độ x, y. Trong khi đó, đối tượng đường, vùng và chuỗi ký tự sẽ là một loạt điểm tọa độ.

### **Cấu trúc Spaghetti**

Với cấu trúcSpaghetti, mỗi điểm được xác định bằng một cặp tọa độ (x,y), mỗi đường được biểu diễn bằng một chuỗi những cặp toạ độ (xi,yi). Mỗi vùng cũng được xác định bởi một cung khép kín và được biểu diễn bằng một chuỗi cặp tọa độ (xi,yi) có tọa độ đầu và tọa độ cuối trùng nhau, nhưng cấu trúc không ghi nhận đặc trưng kề nhau của hai vùng kề nhau, nghĩa là tại đường chung của hai vùng kề nhau có hai đường độc lập. Thí dụ hai vùng 1 và vùng 2 có cung AB chung, cấu trúc Spaghetti biểu diễn hai vùng như sau:

**1**

**2**

A(xA,yA)

B(xA,yA)

*Hình 28. Cấu trúc Spaghetti của vùng 1 và vùng 2*

|  |  |
| --- | --- |
| **Đặc trưng** | **Vị trí** |
| Điểm A | (xA,yA) |
| Cung AB | (xA,yA), (x1,y1), . . . , (xB,yB). |
| Vùng 1 | (x1A,y1A), (x11,y11), . . . , (x1i,y1i), (x1B,y1B), (x1j,y1j), . . . , (x1A,y1A). |
| Vùng 2 | (x2A,y2A), (x21,y21), . . . , (x2i,y2i), (x2B,y2B), (x2j,y2j), . . . , (x2A,y2A). |

Cấu trúc Spaghetti được sử dụng để lập bản đồ số rất tốt, nhưng không thích hợp cho các bài toán phân tích GIS vì không mô tả được các quan hệ không gian.

### **Cấu trúc topology**

Trong GIS, khi biết hình dạng hình học, vị trí, kích thước và hệ tọa độ của đối tượng cũng đủ phục vụ cho việc phân tích không gian nhưng tính topology cũng rất cần thiết. Topology thể hiện mối quan hệ hoặc sự liên kết giữa các đối tượng trong không gian.

Topology là một phương pháp toán học dùng để xác định các quan hệ không gian. Cấu trúc topology còn được gọi là cấu trúc cung-nút (arc-node). Trong cấu trúc này, phần tử cơ bản là cung. Mỗi cung được mô tả như là một chuỗi những đoạn thẳng nối liền nhau, điểm đầu và cuối cung gọi là nút (node), những điểm giữa cung gọi là đỉnh (vertex). Nút là điểm giao nhau của hai hay nhiều cung, đối với những cung độc lập, nút là điểm cuối cùng của cung, không nối liền với bất kỳ cung nào khác. Vùng là một chuỗi những cung nối liền nhau và khép kín, những cung này chính là đường biên của vùng. Một vùng có thể được giới hạn bởi hai đường cong khép kín lồng vào nhau và không cắt nhau.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 70 |  |  |  |  |  |  |  |  |  |
| 60 |  |  |  |  |  |  |  |  |  |
| 50 |  |  |  |  |  |  |  |  |  |
| 40 |  |  |  |  |  |  |  |  |  |
| 30 |  |  |  |  |  |  |  | B |  |
| 20 |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |
| 0 |  |  |  |  |  |  |  |  |  |
|  | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 |

Trục X

a7

A

E

C

E

N1

N2

N3

N4

N5

D

N6

a1

a2

a2

a2

a3

a4

a4

a5

a6

a7

Trục Y

MÃ HOÁ VECTOR TOPOLOGY

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TOPOLOGY CỦA VÙNG | |  | TOPOLOGY CỦA CUNG | | | | |
| Vùng | Đường |  | Đường | Nút đầu | Nút cuối | Vùng trái | Vùng phải |
| A | a1, a5, a3 |  | a1 | N1 | N2 | E | A |
| B | a2, a5, 0, a6, 0, a7 |  | a2 | N2 | N3 | E | B |
| C | a7 |  | a3 | N3 | N1 | E | A |
| D | a6 |  | a4 | N4 | N1 | A | A |
| E | diện tích bên ngoài bản đồ |  | a5 | N3 | N2 | A | B |
|  |  |  | a6 | N5 | N5 | B | B |
|  |  |  | a7 | N6 | N6 | B | C |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TOPOLOGY CỦA NÚT | |  | DỮ LIỆU TOẠ ĐỘ CỦA ĐƯỜNG | | | |
| Nút | Đường |  | Đường | Bắt đầu (x, y) | Trung gian  (x, y) | Kết thúc  (x, y) |
| N1 | a1, a3, a4 |  | a1 | 45,63 | 87, 63 | 87, 53 |
| N2 | a1, a2, a5 |  | a2 | 87, 53 | 87, 9; 11, 9 | 11, 26 |
| N3 | a2, a3, a5 |  | a3 | 11, 26 | 11, 63 | 45, 63 |
| N4 | a4 |  | a4 | 45, 63 | 31, 53 | 31, 42 |
| N5 | a6 |  | a5 | 11, 26 | 27,28;40,35;63,40 | 87, 53 |
| N6 | a7 |  | a6 | 30, 18 |  | 30, 18 |
|  |  |  | a7 | 70, 26 | 73,17;46,17;54,27 | 70, 28 |

*Hình 29. Biểu diễn các đối tượng không gian trong mô hình vector topology*

Trong cấu trúc topology, các đối tượng địa lý được mô tả trong bốn bảng, ba bảng đầu lưu trữ các phần tử không gian vùng, nút, cung, bảng thứ tư lưu trữ tọa độ nút đầu, nút cuối và đỉnh.

* Bảng topology vùng xác định những cung làm đường biên của vùng, phần bên ngoài bản đồ cũng được xem như một vùng không xác định cung đường biên.
* Bảng topology nút xác định mỗi nút thuộc những cung nào.
* Bảng topology cung xác định quan hệ của nút và vùng với cung.

Từ ba bảng này, ta có thể phân tích các quan hệ của các phần tử trong bản đồ. Bảng thứ tư lưu trữ tọa độ của các cung bằng cách lưu trữ tọa độ của các nút và đỉnh của cung, để từ đó vị trí của mỗi phần tử trên bản đồ được liên hệ với thế giới thực. Cấu trúc topology rất thích hợp với những toán tử phân tích không gian, nhất là những bài toán kề (contiguity) và kết nối (connectivity). Trong đó, cấu trúc topology định rõ các liên kết.

*Hình 30. Dạng hình học và Topology của dữ liệu Vector*

Điểm (Point)

Đường (Line)

Chuỗi (String)

Vùng (Area)

Nút (Node)

Chuỗi (chain)

Vùng (Polygon)

(x,y)

(x1,y1), (x2,y2)

{Chỉ số nút,(x,y)}

{Chỉ số chuỗi, chỉ số nút bắt đầu và kết thúc, chỉ số vùng trái và phải}

{Chỉ số vùng, tập các chỉ số chuỗi nằm trong chuỗi theo chiều đồng hồ}

**Bắt đầu**

**Trái**

**Phải**

**Kết thúc**

## 

### **Những đặc điểm topology của cấu trúc vector**

Cấu trúc Topology được dùng để phân tích một mạng bao gồm các nút và các chuỗi, sau đây mô tả một cấu trúc dữ liệu topology :

* Chuỗi bao gồm các trường sau: ID của chuỗi, ID nút bắt đầu, ID nút kết thúc, các thuộc tính.
* Nút bao gồm các trường sau**:** ID của nút, toạ độ (x,y), Các ID của các chuỗi kề nó (dương nếu đi vào nút, âm nếu đi ra khỏi nút )

Bài toán phân tích không chỉ dựa trên mạng mà còn xét đến quan hệ giữa các vùng cần bổ sung có các dạng hình học và topology.

*Hình 32****.*** *Tạo Topology*

b

4

a

d

c

12

11

13

1

3

2

5

6

* Hình học của chuỗi:

|  |  |  |  |
| --- | --- | --- | --- |
| Chuỗi | Bắt đầu | Các tọa độ | Kết thúc |
| 1  2  ...  6 | **(Xa,Ya)**  (Xb,Yb)  (Xc,Yc) | **(X,Y) ...... (X, Y)**  **(X,Y) ...... (X, Y)**  **(X,Y) ...... (X, Y)** | (Xb,Yb)  (Xc,Yc)  (Xc,Yc) |

* Topology của nút

|  |  |
| --- | --- |
| Nút | Chuỗi |
| a  b  c  d | 1, -5, -4  -1, 2, 6  -2, 3, 5  -3, 4, -7 |

* Topology của vùng

|  |  |
| --- | --- |
| Vùng | Chuỗi |
|  | 1, 2, 5  -2, 6, -3  4, -5, 3 |

* Topology của chuỗi

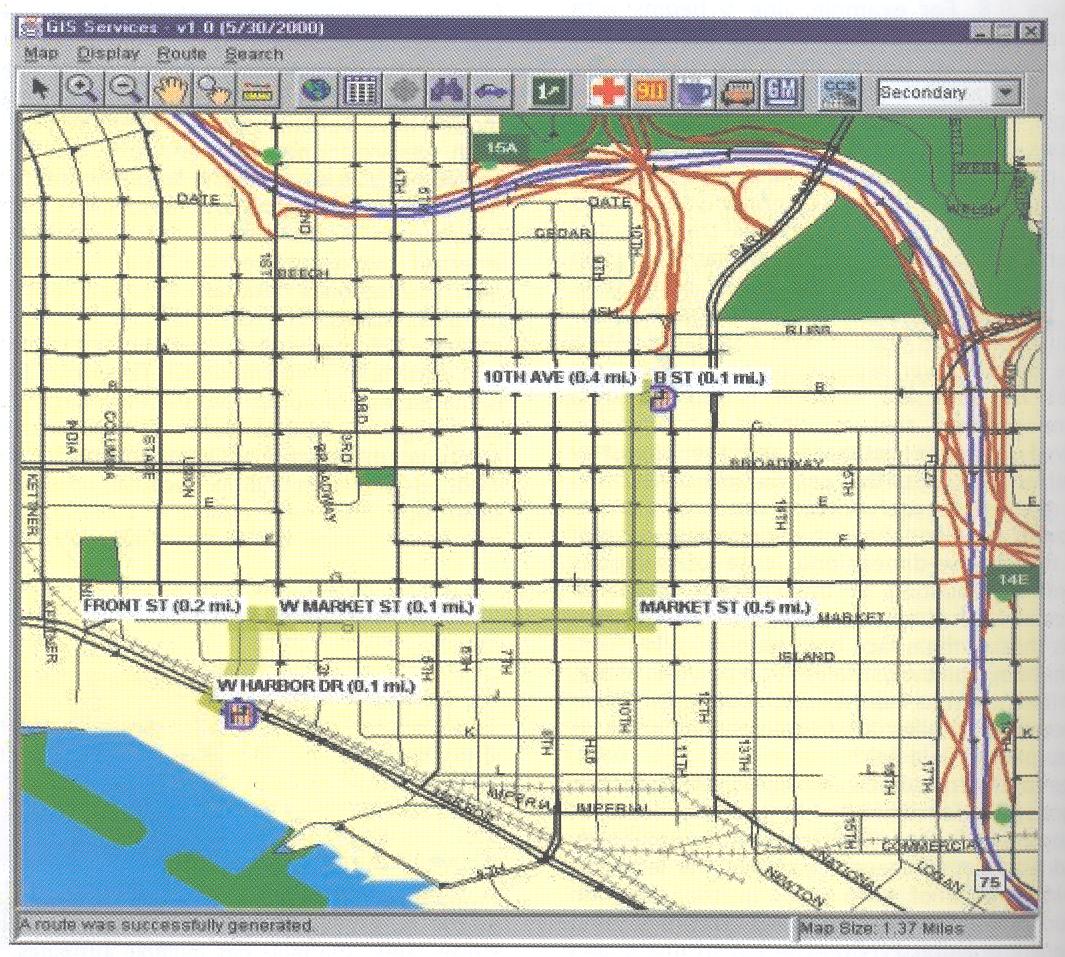
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Chuỗi | Bắt đầu | Kết thúc | Vùng trái | Vùng phải |
| 1  2  ...  6 | a    b  b | b  c  d | 0  .....  0 | +  .... |

Topology của vùng: ID của vùng, Tập ID của các chuỗi.

Topology của chuỗi: ID của chuỗi, ID của nút bắt đầu, ID của nút kết thúc, ID vùng bên trái, ID của vùng bên phải, (các thuộc tính).

## **MÔ HÌNH DỮ LIỆU MẠNG**

Mô hình mạng của GIS là một trường hợp đặc biệt của mô hình dữ liệu topology, được sử dụng để mô hình hóa dòng phân phối hàng hóa. Có hai dạng cơ bản của mô hình dữ liệu mạng GIS: mô hình bán kính (radial) và vòng lặp (looped). Trong mô hình bán kính hay còn gọi là mạng hình cây (tree network) luôn có hướng xuất phát là ngược và xuôi dòng. Việc dự đoán dòng chảy của nước trong mưa bão để xác định vùng ngập lụt là một ví dụ về mô hình mạng hình cây. Trong mô hình mạng vòng lặp sự tự giao nhau xảy ra thường xuyên. Mạng lưới phân phối nước trong thành phố được lặp lại do thiết kế để chắc chắn rằng ảnh hưởng của sự cúp nước cho khách hàng là ít nhất. Trên hình 2.7 trình bày một ví dụ của một mạng đường phố (trích từ [3]). Mạng gồm có một tập hợp các nút (các điểm giao nhau của đường phố) và các đường (đường phố). Mối quan hệ topology giữa các đặc trưng được xác định trong các bảng liên kết. Nhờ thông tin hỗ trợ lưu giữ trong bảng liên kết có thể lần theo dòng xe cộ lưu thông qua mạng và nghiên cứu sự ảnh hưởng của việc tắc đường. Sự cản trở giao thông của những điểm giao nhau và đường phố sẽ xác định tốc độ lưu thông của dòng xe. Rõ ràng, tốc độ của dòng xe sẽ tỷ lệ với giới hạn tốc độ của đường, số làn đường cho xe và thời gian đèn đỏ ở các nút giao thông. Qua ví dụ liên quan đến đường phố, những nguyên tắc cơ bản cũng được áp dụng cho các mạng điện, mạng cung cấp nước thành phố và đường sắt.



***Hình 4.16****. Ví dụ mô hình dữ liệu mạng GIS áp dụng cho giao thông đô thị*

Có rất nhiều ứng dụng sử dụng mô hình dữ liệu mạng như: tìm đường đi ngắn nhất trên mạng lưới giao thông, [xác định vùng dịch vụ](#_Toc76989413), [thiết lập các thuộc tính cho đường đi](#_Toc76989414).

## **MÔ HÌNH DỮ LIỆU** mạng tam giác không đều - **TIN**

Để xây dựng một cấu trúc dữ liệu trong đó chứa đựng yếu tố “độ cao” trên đất liền hoặc “độ sâu” khi ứng dụng cho biển, người ta thường sử dụng khái niệm mô hình số độ cao DEM (Digital Elevation Model) để diễn tả mô hình số độ cao của bề mặt trái đất. Có nhiều thuật ngữ diễn tả khái niệm dữ liệu số địa hình như *Dữ liệu số địa hình* DTD (Digital Terrain Data); *Dữ liệu số địa hình độ cao* DTED (Digital Terrian Elevation Data). DEM có hai mô hình chính: mô hình ma trận độ cao theo lưới ô vuông kiểu GRID và mô hình mạng tam giác không đều TIN (TRIANGULATED IRREGULAR NETWORKS).

1

2

3

4

5

6

7

8

A

B

C

D

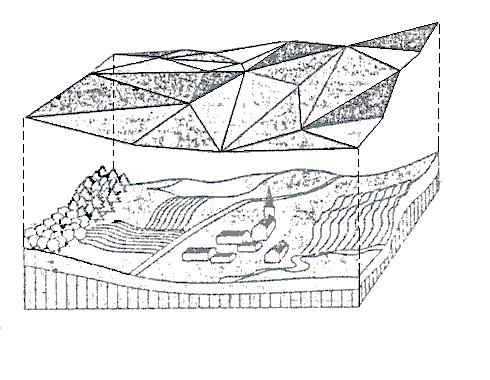
E

F

H

G

|  |  |  |
| --- | --- | --- |
| Tam giác | Nút | Lân cận |
| A | 1, 2, 3 | -, B, D |
| B | 2, 4, 3 | -, C, A |
| C | 4, 8, 3 | -, G, B |
| D | 1, 3, 5 | A, F, E |
| E | 1, 5, 6 | D, H, - |
| F | 3, 7, 5 | G, H, D |
| G | 3, 8, 7 | C, -, F |
| H | 5, 7, 6 | F, -, E |

*a)*

*b)*

***Hình 4.17.*** *Mô hình tôpô của TIN*

Mô hình GID thể hiện quan hệ tôpô giữa các độ cao dạng đối tượng, các điểm độ cao phân bố đều theo lưới ô vuông trong không gian. Thuật toán quản lý mô hình này không phức tạp như mô hình TIN. Một trong những phần mềm quản lý dữ liệu GIS theo mô hình GRID là phần mềm SURFER. Phương pháp GRID phân bố theo lưới không phù hợp với địa hình phức tạp. Để có được độ chính xác mô hình này cần số lượng ghi nhận rất lớn. Mô hình TIN sau đây khắc phục được những nhược điểm trên.

Mô hình TIN gồm một loạt các tam giác không chồng lên nhau, mỗi tam giác xác định một mặt phẳng, các bề mặt này bao phủ lên toàn bộ bề mặt địa hình (Hình 4.17). Các tam giác luôn có 3 nút và thường có 3 tam giác lân cận, riêng tam giác ở các vùng biên của TIN có thể có một hoặc hai tam giác lân cận. Mỗi đỉnh của một tam giác được mã hoá toạ độ cùng với độ cao của nó. Với khái niệm như trên TIN là một mô hình được mô tả ở hình 2.5-D có độ phân giải chi tiết tuỳ thuộc vào độ chính xác của TIN gốc. Khi mở rộng khái niệm ảnh raster 2-D thành không gian 3-D cần phải bổ sung kích thước chiều cao, khi đó các ô pixel phẳng trở thành các khối lập phương và được gọi là *voxels (volume elements).* Hệ tham chiếu địa lý và các thuộc tính có thể gán trong các cấu trúc dữ liệu raster. Khái niệm về voxels rất cần thiết trong nghiên cứu GIS biển và nghề cá.

## **DỮ LIỆU THÔNG TIN QUẢN LÝ DỮ LIỆU (METADATA)**

Để dữ liệu cần được mô tả nguồn gốc và độ chính xác của các lớp GIS để có thể tương thích với nhiều hệ GIS, dữ liệu mô tả này được gọi là *dữ liệu về dữ liệu (metadata)* hay tài liệu về dữ liệu*.* Tập dữ liệu nào không có tài liệu về dữ liệu là không có giá trị. Hiện nay, ta có thể nhận dữ liệu kèm theo báo cáo chất lượng của chúng từ người cung cấp, nó mô tả chính xác nội dung các tệp, thông tin được thu thập từ đâu, cách thức thu thập, dữ liệu đã được kiểm tra ra sao. Nếu không đầy đủ tài liệu về dữ liệu thì phải tìm hiểu để trả lời các câu hỏi sau:

* Dữ liệu có từ bao giờ?
* Dữ liệu xuất xứ từ đâu?
* Phát sinh dữ liệu bằng phương tiện gì?
* Dữ liệu của vùng nào?
* Tỷ lệ bản đồ của dữ liệu?
* Phép chiếu, hệ trục toạ độ, mốc đo đạc sử dụng trong bản đồ?
* Độ chính xác của đặc trưng vị trí và thuộc tính?
* Tính logic và phù hợp của dữ liệu?
* Khuôn mẫu của dữ liệu?
* Dữ liệu được kiểm tra bằng cách nào?

## **ƯU NHƯỢC ĐIỂM CỦA CẤU TRÚC DỮ LIỆU RASTER VÀ VECTOR**

Hiện nay trên thị trường đa số các phần mềm GIS sử dụng dạng mô hình dữ liệu vector. Tuy nhiên, các phần mềm đều có các công cụ hỗ trợ để tích hợp cả hai loại dữ liệu hoặc chuyển đổi qua lại với nhau. Thông thường là sự tích hợp dữ liệu raster và vector, như sự tích hợp các ảnh viễn thám vào trong một lớp bản đồ vector để phục vụ mục tiêu nghiên cứu.

Tóm tắt các ưu nhược điểm của hai loại mô hình như sau:

##### Mô hình Raster

*Ưu điểm:*

1. Có cấu trúc dữ liệu đơn giản.
2. Các thao tác chồng lớp dễ dàng và hiệu quả.
3. Có thể sử dụng công nghệ quét ảnh để số hoá bản đồ nên giá thành thấp.
4. Các kỹ thuật xử lý ảnh cho ra dữ liệu để tích hợp GIS ở dạng raster.
5. Phân tích diện tích và đa giác đơn giản.
6. Hợp nhất và chồng lớp đơn giản.
7. Kỹ thuật rẻ và ngày càng có lợi thế trong tương lai.
8. Thích hợp tốt để chia nhỏ các thay đổi liên tục trong không gian.

*Nhược điểm:*

1. Dung lượng lưu trữ và xử lý dữ liệu lớn.
2. Với độ phân giải thấp sẽ mất độ chi tiết.
3. Bản đồ cuối cùng thường ở dạng thô, đặc biệt ở những phần mềm rẻ tiền.
4. Phân tích tuyến tính khó khăn.
5. Khó thể hiện các mối quan hệ tôpô.

##### Mô hình vector

*Ưu điểm:*

1. Dữ liệu được nén một cách tương đối, vì vậy yêu cầu dung lượng lưu trữ thấp.
2. Các đặc tính có thể định vị chính xác.
3. Tôpô có thể mô tả hoàn toàn với liên kết mạng.
4. Các nét rất nhỏ có thể được hiển thị và tất cả các nét có thể được vẽ một cách chính xác.
5. Dữ liệu về các đặc tính có thể gọi ra dễ dàng để cập nhật và hiệu chỉnh.
6. Phân tích tuyến tính dễ dàng.

*Nhược điểm:*

1. Cấu trúc dữ liệu phức tạp.
2. Thao thác chồng lớp bản đồ khó khăn.
3. Thể hiện sự biến đổi không gian cao không có hiệu quả.
4. Không thể thực hiện thao tác xử lý và nâng cao các ảnh số có hiệu quả.
5. Truy xuất dữ liệu chậm.
6. Các phân tích vùng và đa giác khó khăn.
7. Tốn kém cho việc tạo dữ liệu và cả bản quyền phần mềm.

# BÀI TẬP CHƯƠNG 2

1. Trình bày mô hình dữ liệu Vector, ưu nhược điểm của dữ liệu Vector ?
2. Trình bày mô hình dữ liệu Raster, ưu nhược điểm của dữ liệu Raster ?
3. So sánh mô hình raster và với vector (ưu, nhược điểm)

# 

# ***CHƯƠNG 3*** THU THẬP THIẾT KẾ VÀ XÂY DỰNG DỮ LIỆU GIS

## **THU THẬP DỮ LIỆU KHÔNG GIAN**

Để GIS hoạt động có hiệu quả các dữ liệu thu thập sau khi thu thập vào được xử lý sơ bộ trước khi thực hiện các thao tác trên chúng.

Thu thập số liệu được thực hiện bằng nhiều phương pháp khác nhau từ các nguồn như bảng biểu, từ các tài liệu có sẵn, ảnh tư liệu nhờ các thiết bị không ảnh, số hóa từ các bản đồ có sẵn thông qua kết quả đo trực tiếp hay xử lý số liệu có sẵn.

### **Thu thập dữ liệu**

Để các thao tác phân tích trên dữ liệu địa lý cho kết quả có chất lượng cao, dữ liệu phải được thu thập, nhận diện và đo đạc vào cùng thời điểm, cùng độ phân giải không gian, sử dụng cùng một thủ tục nhận diện và được nhập vào hệ thống GIS theo cùng một phương pháp. Tuy nhiên, trong thực tế việc thu thập dữ liệu lại được thực hiện vào những thời điểm khác nhau và thường có độ phân giải khác nhau. Có dữ liệu được thu thập bằng cách đo đạc trực tiếp trong khi dữ liệu được thu thập từ bản đồ có sẵn. Nhưng trong mọi trường hợp việc thu thập tọa độ vị trí các đối tượng và thu thập các thuộc tính của chúng phải tiến hành vào cùng thời điểm.

Nhập dữ liệu là công việc tốn kém thời gian nhất trong thao tác GIS, thường tốn đến 80% ngân sách của một dự án. Vì thế, người ta thường sử dụng tối đa nguồn dữ liệu có sẵn từ các công ty hay tổ chức khác.

Có hai nhóm phương pháp thu thập dữ liệu. Nhóm thứ nhất là thu thập dữ liệu từ chính các đối tượng, phương pháp này cho kết quả chính xác nhưng chi phí cao. Đó là:

* Trắc địa mặt đất.
* Phương pháp định vị bằng vệ tinh (GPS).
* Chụp ảnh bằng máy bay hay vệ tinh.

Nhóm thứ hai là thu thập dữ liệu từ các nguồn số hóa hay tương tự có sẵn, dữ liệu thu thập được từ nhóm phương pháp này ít chính xác, chi phí thấp. Đó là:

* Số hoá bằng tay các bản đồ giấy.
* Số hoá tự động bản đồ giấy bằng máy quét.
* Sử dụng các CSDL bản đồ số hóa có sẵn.

### **So sánh các nguồn thu thập dữ liệu**

Dữ liệu GIS có thể thu thập từ nhiều nguồn khác nhau, nhiều dạng khác nhau và được lưu dưới nhiều phương pháp khác nhau.

Các phương pháp thu thập dữ liệu

Chuyển   
đổi

Số hóa

Ảnh

Quét

Viễn thám

Toàn đạt điện tử

GPS

## **THIẾT KẾ DỮ LIỆU KHÔNG GIAN**

Cơ sở dữ liệu GIS gồm hai phần: dữ liệu không gian (thông tin địa lý/thông tin hình học) của các đối tượng và dữ liệu thuộc tính (thông tin phi không gian/ thông tin phi địa lý). Giữa dữ liệu không gian và dữ liệu thuộc tính phải được liên kết với nhau. Một đối tượng trong bản đồ sẽ liên kết với một bản ghi trong dữ liệu thuộc tính. Do đó khi tìm kiếm được một đối tượng không gian trên bản đồ ta có thể xem ngay dữ liệu thuộc tính tương ứng với nó ngược lại thao tác truy vấn theo dữ liệu thuộc tính, ta cũng có thể tìm được đối tượng không gian tương ứng trên bản đồ. Quan hệ giữa dữ liệu không gian và dữ liệu thuộc tính có thể được mô tả như hình sau:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | TEN | SH | LOAI |
| 01 | P1 |  |  |
| 02 | A1 | Thu | Tôm |
| 03 | A2 | Hồng | Rau |
| 04 | L1 |  | Đường |

Dữ liệu không gian là biểu diễn hình học của các đối tượng địa lý liên kết với vị trí trên thế giới thực. Những đối tượng địa lý có thể được biểu diễn dưới dạng điểm, đường hoặc vùng.

Dữ liệu thuộc tính là những thuộc tính mô tả đặc điểm của các đối tượng địa lý.

## **Dữ liệu không gian**

Dữ liệu dạng điểm dùng để biểu diễn các đối tượng không gian quá nhỏ, không thể diễn giải bằng một đường hoặc một vùng. Điểm được đại diện để xác định vị trí của đối tượng như điểm dân cư, vị trí thành phố, nhà máy, xí nghiệp, trường học,…

Dữ liệu không gian dạng đường dùng để biểu diễn các đối tượng có dạng tuyến, dài và hẹp, đối tượng không thể thể hiện như một vùng, ví dụ như đường giao thông, đường ranh giới, sông suối,…

Vùng dùng để biểu diễn các đối tượng có bề mặt đồng nhất như vùng hành chính quốc gia, tỉnh, huyện, xã, vùng núi, vùng biển, vùng sản xuất nông nghiệp, lâm nghiệp,…

Tùy thuộc vào mục đích sử dụng mà ta có thể quyết định đối tượng được biểu diễn dưới dạng điểm, đường hay vùng.

Kết quả của việc thiết kế dữ liệu không gian là danh sách các lớp dữ liệu nhằm đáp ứng yêu cầu sử dụng.

## **Dữ liệu thuộc tính**

Trong GIS sử dụng phổ biến CSDL thuộc tính ở dạng bảng, là mô hình quản lý dữ liệu theo kiểu sắp xếp thành bảng theo dòng và cột. Mỗi cột là một trường (field) và mỗi dòng là một biểu ghi (record). Các trường gồm các kiểu dữ liệu:

Text – Văn bản

Number - Số, gồm có byte, số nguyên (integer), số thập phân (decimal), số thực (float).

Date – Ngày, tháng.

Time - Thời gian

Logical - Trường logic (True/ False).

Trong trường văn bản có thể tạo các chỉ thị (index), thường là mã các đối tượng quản lý nhằm để nâng cao tốc độ tìm kiếm của máy tính. Giống như các CSDL kiểu quan hệ khác, thông tin thuộc tính của GIS có đầy đủ các toán tử tính toán, toán tử quan hệ, các hàm về chuỗi, các hàm thống kê, các câu lệnh của ngôn ngữ truy vấn SQL v.v. Ngoài ra, chúng còn các phép tính về hình học như xác định diện tích, chu vi, chiều dài, tọa độ trọng tâm của các đối tượng hình học, v.v ...

Dữ liệu thuộc tính của các đối tượng không gian được tổ chức thành cơ sở dữ liệu và được quản lý bởi một hệ quản trị cơ sở dữ liệu (DBMS). Cơ sở dữ liệu thuộc tính được xây dựng theo các mô hình: mô hình phân cấp, mô hình mạng hay mô hình quan hệ. Trong đó, mô hình quan hệ được sử dụng phổ biến nhất. Tuy nhiên hiện nay, nhiều cơ sở dữ liệu thuộc tính phi không gian trong các hệ thống thông tin địa lý cũng đã bắt đầu được xây dựng theo mô hình đối tượng.

## **SỐ HÓA BẢN ĐỒ**

Bản đồ được thành lập theo công nghệ truyền thống thể hiện nội dung lên mặt phẳng thông dụng là mặt phẳng giấy. Sản phẩm dạng này ta gọi là phiên bản cứng (hard copy). Nhiệm vụ của công nghệ thành lập bản đồ số từ phiên bản cứng là biến đổi phiên bản cứng thành phiên bản mềm (soft copy). Quá trình này gọi tắt là số hóa bản đồ, là tiến trình chuyển đổi các đối tượng trên bản đồ giấy sang dạng số.

Ta có thể số hoá bản đồ giấy bằng bàn số hóa hoặc nhập bản đồ bằng máy quét. Số hoá bằng bàn số hóa là tiến trình ghi lại vị trí của trình tự các điểm đặc trưng dọc theo đường trên bản đồ. Phương pháp này cho kết quả dưới dạng bản đồ vectơ 2D. Còn kết quả của việc nhập bản đồ bằng máy quét là số liệu raster. Nếu sử dụng công cụ phần mềm chuyển đổi raster sang vectơ ta cũng có số liệu bản đồ vectơ 2D. Một trong các đặc trưng quan trọng của dữ liệu bản đồ là độ chính xác phụ thuộc vào độ chính xác của bản đồ giấy và quá trình số hoá hay quét bản đồ.

Khi số hoá một bản đồ, ta sử dụng một bàn số hoá đã được kết nối với máy tính để đồ lại các đối tượng muốn số hóa. Các toạ độ x, y của đối tượng đó được ghi lại một cách tự động và được lưu trữ như dữ liệu không gian. Ta có thể tạo ra một lớp dữ liệu hoàn toàn mới, hay có thể cập nhật dữ liệu cho một lớp dữ liệu đang tồn tại.

Bản đồ dùng để số hóa phải rõ ràng, dữ liệu được cập nhật, phẳng và không nhàu nát hoặc bị co giãn do thời tiết thay đổi.

Việc số hóa phải đảm bảo các yêu cầu sau:

* Đảm bảo các liên kết tồn tại giữa các điểm.
* Các đường song song vẫn còn song song.
* Bảo toàn các vị trí tương đối.
* Bảo toàn các vị trí tuyệt đối (thông qua toạ độ)
* Bảo toàn tính liền kề.
* Các đường gần sát nhau không được cắt nhau.

Trước khi số hóa các đối tượng trên bản đồ, cần phải cố định bản đồ gốc, gọi là đăng ký toạ độ của bản đồ bằng cách khai báo tọa độ thực của các điểm khống chế.

### **Tỉ lệ bản đồ**

Tỉ lệ bản đồ là mức độ thu nhỏ đối tượng trên mặt đất khi biểu diễn trên bản đồ.

Thí dụ, nếu chúng ta vẽ một con đường dài 4.8 km trên thực tế thành một đường dài 20 cm trên bản đồ, tương ứng với tỉ lệ 20 cm: 4.8 km ~ 20 cm: 480.000 cm. Vậy tỉ lệ của bản đồ là 1:24.000 (1 cm trên bản đồ tương ứng với 24.000 cm ngoài thực tế).

Tỉ lệ bản đồ được thể hiện bằng nhiều cách: dạng số (1:10.000), dạng chữ (1 cm trên bảng đồ tương ứng 10.000 cm ngoài thực địa), dạng thước.

Kích thước của những đối tượng trên các bản đồ tỉ lệ nhỏ (1:1.000.000; 1:500.000; …) sẽ nhỏ hơn các đối tựơng trên bản đồ tỉ lệ lớn (1:2.000; 1:1.000; 1:500; …). Một cách tổng quát, những bản đồ tỉ lệ nhỏ biểu diễn những vùng rộng lớn trên mặt đất nhưng co sự phân giải không gian thấp, vì vậy biểu diễn ít chi tiết. Ngược lại, những bản đồ tỉ lệ lớn biểu diễn những vùng mặt đất nhỏ nhưng có độ phân giải không gian cao, vì vậy biểu diễn chi tiết hơn.

### **Chọn lưới chiếu bản đồ**

Lưới chiếu bản đồ là một phép toán học để chuyển đổi bề mặt cong của trái đất thành mặt phẳng bản đồ. Không co một mặt phẳng nào biểu diễn trái đất được thiết kế hoàn toàn chính xác, do đó có nhiều loại lưới chiếu khác nhau ra đởi phục vụ cho nhiều mục đích khác nhau.

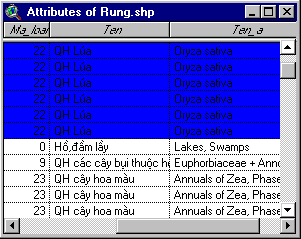
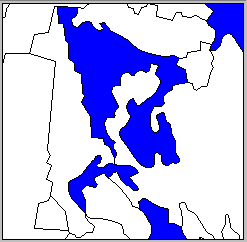
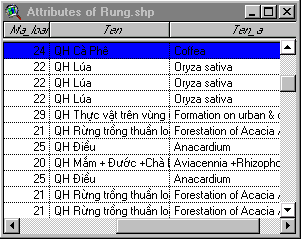
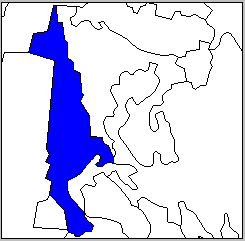
Mỗi lưới chiếu có thể xác định cho một loại thuộc tính của bề mặt như lưới chiếu đồng góc, lưới chiếu đồng diện tích, v.v…

**Những vấn đề cần xem xét khi chọn lưới chiếu**

* Những thuộc tính không gian nào cần bảo tồn.
* Dữ liệu không gian nằm ở khu vực nào: vùng cực hay vùng xích đạo.
* Hình dáng của dữ liệu:hình vuông hay trải rộng theo hướng đông-tây.
* Diện tích bao phủ của bản đồ lớn bao nhiêu?

# ***CHƯƠNG 4*** TRUY VẤN PHÂN TÍCH VÀ HIỂN THỊ DỮ LIỆU

Truy vấn dữ liệu đề cập đến việc tìm kiếm các đối tượng dữ liệu không gian (và các thuộc tính của chúng) hiện diện tại một vị trí cụ thể hoặc tìm kiếm các vị trí có những đối tượng không gian thoả mãn một tiêu chuẩn cụ thể. Khi các truy vấn được thực hiện một cách tương tác với nhau thì các kết quả thường được biểu diễn dưới dạng các khung nhìn được liên kết động.



Truy vấn dữ liệu từ một cơ sở dữ liệu GIS là trả lời một trong hai câu hỏi sau:

* Câu hỏi một: Từ những dữ liệu không gian được hiển thị trên màn hình, hãy cho biết những đặc tính của một đối tượng được chọn và hiển thị những đặc tính đó trên màn hình?
* Câu hỏi hai: Từ các bảng dữ liệu thuộc tính được hiển thị trên màn hình, hãy tìm đối tượng không gian tương ứng và hiển thị đối tượng này lên màn hình?

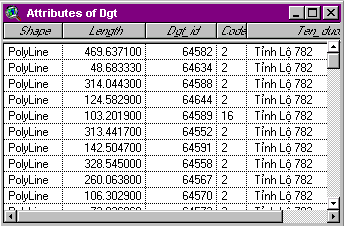
Trong câu hỏi một, người sử dụng GIS dùng con trỏ chỉ vào một đối tượng bất kỳ trên dữ liệu không gian được hiển thị trên màn hình, kích vào đó để truy xuất dữ liệu thuộc tính của đối tượng mà con trỏ chỉ vào.

Trong câu hỏi hai, người sử dụng nhập những thuộc tính đã biết như: “Quốc lộ 1”, “mật độ dân cư > 5000 người/km2”, “vùng ô nhiễm bụi”, “vùng trồng lúa”, …. để tìm vị trí của các đối tượng trên dữ liệu không gian thoả điều kiện nhập vào. Các đối tượng được chọn có thể được đánh dấu trên màn hình hay được thể hiện trong một cửa sổ khác.

## **TRUY VẤN TỪ DỮ LIỆU THUỘC TÍNH**

Đối với hệ thống thông tin địa lý, để xây dựng một cơ sở dữ liệu từ không gian thực, chúng ta phải mô tả dữ liệu dưới dạng số hóa một cách có phương pháp và định lượng với số liệu thống kê được từ thế giới thực. Trong đó, dữ liệu thuộc tính thiết lập mối quan hệ, liên kết đến những bản đồ với những số liệu thống kê mà ta đã mô tả và sau đó tự động hóa những đặc tính địa lý và kết quả đặt trên bản đồ để xử lý.

Trong hệ thống thông tin địa lý, tất cả dữ liệu thuộc tính được thiết lập thành một cấu trúc trong một bảng. Những dòng của bảng là các mẫu tin (record) và những cột của bảng là các thuộc tính (attribute). Mỗi mẫu tin đi với một thuộc tính chứa đựng một giá trị và giá trị đó được chia thành nhiều kiểu dữ liệu như: chữ (text), số (number), luận lý (boolean), ngày (date), … dựa trên các giá trị này chúng ta có thể tìm kiếm, sửa đổi hoặc cập nhật các thông tin mới.



Hầu hết hệ thống thông tin địa lý đều được xây dựng một hệ quản trị cơ sở dữ liệu khá căn bản để quản lý dữ liệu địa lý hoặc dựa vào khả năng của một hệ quản trị cơ sở dữ liệu (DBMS) hiện có. Khi đó, việc truy vấn dữ liệu được điều khiển bởi khả năng của người quản lý cơ sở dữ liệu. Tất cả những hệ quản trị cơ sở dữ liệu bao gồm những chức năng về hiển thị dữ liệu căn bản, đó là hiển thị tất cả các thuộc tính trong một cơ sở dữ liệu, hiển thị tất cả các mẫu tin cùng với các thuộc tính của chúng và hiển thị tất cả các cơ sở dữ liệu đang tồn tại. Đồng thời, hệ quản trị cơ sở dữ liệu cũng cho phép xuất các mẫu tin theo một định dạng chuẩn, với cách trình bày trang đặc thù.

Trên thực tế, việc tìm kiếm thông tin trong cơ sở dữ liệu là đáng quan tâm nhất, hệ quản trị cơ sở dữ liệu phải hỗ trợ các chức năng thuộc về phạm trù *truy vấn*. Hệ quản trị cơ sở dữ liệu phải cho phép truy vấn dữ liệu tại bất kỳ mẫu tin và bất kỳ yêu cầu truy vấn nào.

*Find* là dạng tìm kiếm từ dữ liệu thuộc tính cơ bản nhất. *Find* thường được dùng để tìm kiếm trên từng mẫu tin riêng lẻ, có thể thông qua loại tìm kiếm bằng *search* (dò tìm) hoặc *browse* (duyệt qua). *Browse* dò tìm qua từng mẫu tin cho đến khi tìm thấy một mẫu tin cần tìm.

Một cách tìm kiếm trên thuộc tính khác là tìm kiếm bằng *sorting* (phân loại), việc phân loại này được thực hiện bằng cách cách sắp xếp thứ tự các giá trị hiện có trong một trường thuộc tính, nếu trường thuộc tính kiểu chữ thì các thuộc tính được sắp xếp theo thứ tự abc, nếu trường thuộc tính kiểu số thì các thuộc tính được sắp xếp theo thứ tự giá trị số tăng dần hoặc giảm dần. Lưu ý rằng, việc sắp xếp các giá trị không quan tâm đến các giá trị khuyết và vị trí đặt các giá trị.

Phép toán *restrict* (giới hạn) cho phép người sử dụng gọi ra một tập con trong tổng số các mẫu tin bằng cách đặt ra giới hạn trên các giá trị thuộc tính. Thí dụ, chúng ta có thể gọi ra tất cả các mẫu tin có ngày cập nhật sau ngày 1/1/99, hoặc các thành phố có dân số hơn 100.000 người.

Phép toán *select* (chọn lọc) cho phép chúng ta chọn lựa các thuộc tính sẽ được rút ra từ một cơ sở dữ liệu đưa vào một cơ sở dữ liệu mới với ít thuộc tính hơn, công việc này thường được thực hiện bằng cách kết nối các mẫu tin và các thuộc tính của một cơ sở dữ liệu vào một cơ sở dữ liệu khác có liên quan.

Phép toán *compute* (tính toán) cho phép tính toán một giá trị cho một thuộc tính, để ấn định giá trị cho thuộc tính hoặc làm các phép toán cộng, trừ, nhân, chia giữa các thuộc tính với nhau. Chúng ta cũng có thể sử dụng phép toán *renumber* (đánh số lại) cho một thuộc tính, tức là thay đổi các giá trị thuộc tính theo chỉ định. Thí dụ, chúng ta có thể muốn tìm tất cả những giá trị phần trăm trong một thuộc tính nhỏ hơn 50% và gán giá trị là 0 và những giá trị lớn hơn 50% được gán giá trị là 1, sau đó có thể tổ hợp nhị phân với các thuộc tính khác.

Trong việc thay đổi giá trị này, lưu ý rằng các lệnh đã quy định một dòng tại một thời điểm và thường phải được sử dụng kết hợp để đạt được kết quả mong muốn. Cũng lưu ý rằng, hầu hết các hệ cơ sở dữ liệu thực hiện các chức năng khai thác cơ sở dữ liệu trên một tập dữ liệu tạm của cơ sở dữ liệu, khi cần thiết mới lưu trở lại vào cơ sở dữ liệu chính thức. Nhiều hệ quản trị cơ sở dữ liệu sử dụng các menu, các ngôn ngữ truy vấn, từ khóa, câu lệnh để đạt được nhiều kết quả cùng một lúc.

## **TRUY VẤN TỪ DỮ LIỆU KHÔNG GIAN**

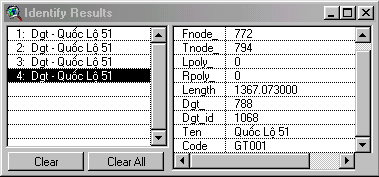
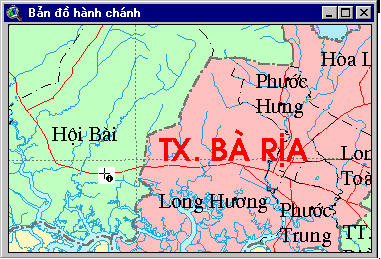
Khi nghiên cứu nội dung truy vấn từ thuộc tính, chúng ta đã xem xét đến các lệnh tìm kiếm và khôi phục dữ liệu như: *find, browse, sort, restrict, renumber* và *compute*. Chuyển sang truy vấn trên dữ liệu không gian trong hệ thống thông tin địa lý, một số phép toán có thể chỉ là ước lượng không gian, nhưng cũng có một số phép toán phức tạp hơn.

Trong dữ liệu không gian thì các mẫu tin được thay bằng các đối tượng địa lý. Có một số thuộc tính riêng biệt đặc trưng cho dữ liệu không gian, các thuộc tính đó có liên quan đến các tọa độ, phạm vi không gian cộng với các đặc điểm của các đường và các vùng. Khi đó, việc hiển thị các đối tượng không gian sẽ bao gồm hiển thị tọa độ thực của các đối tượng, chiều dài của các cung và diện tích của các vùng. Áp dụng các chức năng tìm kiếm từ thuộc tính đối với các đặc điểm của đối tượng không gian để có kết quả trên dữ liệu không gian. Thí dụ, chúng ta có thể tìm tất cả các cung có chiều dài lớn hơn chiều dài của một cung nào đó hoặc các vùng có diện tích lớn hơn 1 hecta.

Việc hiển thị tất cả các mẫu tin trong một không gian xảy ra hai trường hợp: hoặc hiển thị tất cả các thuộc tính hoặc hiển thị tất cả các đối tượng trên một bản đồ. Tạo ra một bản đồ để cho phép chúng ta tìm kiếm thông tin bằng mắt là một phép toán truy tìm theo không gian cũng được GIS quan tâm.

Một cách truy vấn trên không gian đơn giản và thuận tiện nhất đó là sử dụng *identify* (nhận dạng) hoặc *locate* (định vị), cách này sử dụng con trỏ chuột chỉ vào một đối tượng không gian, lúc đó các thông tin thuộc tính của đối tượng sẽ được hiển thị trong bảng kết quả (hình 5.3).

Việc tìm kiếm từ không gian theo cách duyệt qua bản đồ và chọn đối tượng là một khả năng rất mạnh của GIS, đặc biệt nếu hệ thống thông tin địa lý được đặt trên một máy tính xách tay, đối tượng được định vị là nơi bạn đang đứng và vị trí đó được xác định thông qua máy thu GPS. Trên bản đồ cho phép tìm kiếm bằng cách chỉ định một đối tượng riêng lẻ, tất cả các đối tượng trong một hình chữ nhật mà chúng ta kéo chuột để vẽ ra hoặc tất cả các đối tượng trong một vùng bất kỳ được vẽ trên màn hình bằng công cụ vẽ bản đồ.



Phân loại đối tượng là cách tìm kiếm ít có ý nghĩa trong truy vấn từ không gian, thường được sử dụng để đánh giá xu hướng của các đối tượng không gian thu được từ việc phân loại thuộc tính. Thí dụ, chúng ta có thể phân loại các vùng hành chính huyện theo mật độ dân số (hình 5.4), phân loại các vùng trồng trọt theo 3 mức: cao, trung bình và thấp của mật độ cây trồng, …

Phép toán *select* trong truy vấn theo không gian nhằm rút ra các thuộc tính đặc trưng. Nhờ vậy mà chúng ta có thể chỉ sử dụng các lớp dữ liệu chứa các đối tượng cần thiết cho công việc. Việc chọn lọc để lấy các sông chính từ lớp dữ liệu mạng lưới sông ngòi để đưa vào một lớp mới chỉ chứa các sông chính là một thí dụ của phép toán *select* trong truy vấn từ không gian.

Một kiểu chọn lọc đối tượng không gian thường được sử dụng nhất trong các phép toán GIS đó là phép toán *buffer* (vùng đệm). Phép toán này cho phép chọn những đối tượng nằm trong một khoảng cách xác định tính từ một điểm, một tập các điểm, một đường hoặc một vùng. Các *buffer* bao quanh một điểm có dạng vùng hình tròn, quanh một đường có dạng vùng ngoằn ngoèo và quanh một vùng có dạng vùng rộng lớn hơn (hình 5.5).

*a) Buffer điểm*

*Hình 39.**Các loại vùng đệm*

*b) Buffer đường*

*c) Buffer vùng*

Chúng ta có thể sử dụng phép toán *buffer* để giải quyết nhiều bài toán dạng như: tìm những căn nhà vi phạm lộ giới, tìm những trạm y tế cách khu vực có dịch bệnh trong vòng bán kính 5 km, khoanh vành đai bảo vệ công trình ngầm, …

Đối với dữ liệu thuộc tính, phép toán kết hợp được thực hiện bằng cách hợp nhất các thuộc tính thành các tập tin phẳng. Trong không gian, việc kết hợp này được gọi là chồng lớp bản đồ (map overlay). Trong chồng lớp bản đồ, một bản đồ mới được tạo ra dựa trên sự phân chia không gian của cả hai bản đồ gốc (hình 5.6). Mỗi vùng được tạo ra trên lớp bản đồ mới có một mẫu tin thuộc tính mới trong bảng thuộc tính không gian và mang các thuộc tính của cả hai bản đồ gốc.

Một chức năng nữa trong truy vấn không gian được xem là hữu ích nhất đó là tìm kiếm và kiểm tra thông qua các mối quan hệ của các điểm, đường và vùng. Sự kết hợp của các đối tượng cho phép chúng ta ấn định các thuộc tính từ các đối tượng điểm đến các đối tượng vùng. Có ba kiểu tiêu biểu trong chức năng này là tìm các điểm trong vùng, các đường trong vùng và điểm trong một cách đến một đường. Thí dụ, tìm các khách sạn trong địa bàn Quận 5, tìm các đường dây tải điện thuộc địa bàn quản lý của điện lực Chợ Lớn hoặc tìm các bể chứa dầu nằm trong khoảng cách 3km so với các con sông (dạng đường), …

*Hình 40.* *Chồng lớp hai lớp dữ liệu dạng vùng A và B để tạo ra lớp dữ liệu C chứa cả A và B.*

A1

A2

B1

B2

B3

B4

C1

Cuối cùng, một số phép tính toán nâng cao sử dụng trong truy vấn không gian như các phép tính toán line-of-sight, tính toán trên một khu vực được nhìn thấy từ một vị trí trên bản đồ biểu thị hình dáng của địa hình: như các bản đồ hiển thị độ dốc và hướng của độ dốc, có thể được dùng để ước tính khả năng ngập lụt; ước tính lưu lượng vận chuyển trên mạng đường giao thông để dự báo nạn kẹt xe; …

## **PHÂN TÍCH KHÔNG GIAN**

Trong hầu hết các trường hợp, các truy vấn thuộc tính và các truy vấn không gian có thể được đưa ra với một hệ GIS khi câu hỏi đòi hỏi không có sự thay đổi dữ liệu. Tuy nhiên, một phân tích không gian có thể yêu cầu sự hình thành dữ liệu mới từ dữ liệu gốc. Ví du, giả sử chúng ta muốn tìm mối liên hệ giữa mức độ ồn và khoảng cách gần đường cao tốc (ví dụ trong phạm vi một dặm của đường cao tốc). Để trả lời câu hỏi này, bạn phải kết hợp (ví dụ: chồng lớp (overlay)) bản đồ rảnh thửa, bản đồ đường cao tốc và các khu vực trong phạm vi 1 dặm của đường cao tốc nhận được từ bản đồ đường cao tốc. Quá trình này có thể yêu cầu sự mô tả các thực thể địa lý mới và tạo ra bảng dữ liệu mới cho thấy sự kết hợp của một số nhân tố.

Một số phép phân tích không gian: phân tích chồng lớp, phân tích bề mặt, phân tích mạng…

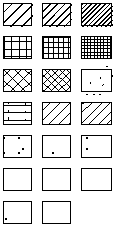
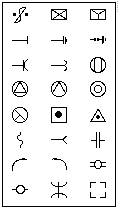
## **KÝ HIỆU HÓA VÀ TẠO TRANG IN BẢN ĐỒ**

### **Ký hiệu hóa**

Một ảnh bản đồ bao gồm hai thành phần chính đó là các yếu tố đồ họa được liên kết với các đối tượng không gian và các chú thích bản đồ.

Mỗi loại đối tượng không gian như các điểm, đường, vùng, vùng không đều và các ảnh điểm, có thể được liên kết đến các yếu tố đồ họa để hiển thị.

* *Dữ liệu điểm* được biểu diễn trên các ảnh bằng các ký hiệu đồ họa. Kiểu mẫu, kích thước và màu sắc của ký hiệu được hiển thị tại vị trí của điểm, chúng có thể là tập hợp các giá trị hằng số hoặc có thể biến thiên theo một hoặc nhiều trường thuộc tính trong bảng thuộc tính điểm. Hình 41.a minh họa một số kiểu ký hiệu dạng điểm.
* *Dữ liệu đường* được biểu diễn trên các ảnh bằng các đường liên tục, các đường đó có thể thay đổi theo độ dày mỏng của đường, theo màu sắc hoặc theo kiểu đường. Hình 41.b ??? minh họa một số kiểu ký hiệu dạng đường.
* *Dữ liệu vùng không đều* được biểu diễn trên ảnh như những vùng khép kín được tô bằng màu hoặc trắng đen, hoặc các kiểu mẫu nền (hình 7.1 minh họa một số kiểu ký hiệu mẫu nền). Đường biên của vùng có thể biểu diễn dạng đường, với kiểu mẫu, lực nét, màu sắc tương tự như cách biểu diễn dữ liệu dạng đường. Màu sắc và kiểu mẫu của vùng kín được xác định bằng các giá trị trong một hoặc nhiều trường dữ liệu của bảng thuộc tính vùng và sự phân loại thường được áp dụng để phân chia các thuộc tính của chúng thành các lớp riêng biệt. Các màu sắc được xác định sử dụng một bảng màu hay bảng tra màu (LUT). Các ký hiệu trên mẫu nền có thể là rời rạc hoặc liên tục, có thể thay đổi mật độ.
* *Dữ liệu ảnh điểm* được biểu diễn hầu hết chỉ bằng màu sắc hoặc các mức xám. Trong trường hợp một ảnh raster không được liên kết với bảng thuộc tính, màu sắc được xác định từ giá trị ảnh điểm sử dụng một bảng tra màu LUT. Mặt khác, giá trị ảnh điểm có thể chỉ đến một bảng thuộc tính trong trường hợp giá trị của trường dữ liệu đã chọn được sử dụng để xác định màu sắc.



*a. Chú thích bản đồ*

Chú thích thường được tạo ra sau cùng, đặt lên trên các yếu tố đồ họa của ảnh. Chú thích bao gồm nhãn, tiêu đề, bảng chú giải, thước tỉ lệ, lưới chiếu và mũi tên chỉ hướng Bắc. Các nhãn thường được liên kết đến các đối tượng điểm để chúng có thể được đặt tại những vị trí riêng biệt trên bản đồ. Các trường dữ liệu trong bảng thuộc tính có thể được sử dụng để xác định chuỗi ký tự, kiểu chữ, kích thước, màu sắc và hướng của các nhãn. Trong một số hệ thống, kích thước của nhãn có thể tự động thay đổi theo sự tăng, giảm của tỉ lệ bản đồ. Tiêu đề và các khối văn bản cũng có thể được định vị tại các vị trí xác định và cũng có các đặc tính tương tự như các nhãn. Các mắc lưới hay các đường kẻ chia độ bao quanh hoặc cắt ngang qua bản đồ được sử dụng để biểu thị lưới chiếu và các tọa độ địa lý. Trong một vài hệ thống, chú giải được liên kết đến các lớp đối tượng không gian.

Nhìn chung, việc thiết kế chú thích bản đồ là một việc làm linh hoạt và có tính tương tác lẫn nhau, sao cho các bản đồ tạo ra có chất lượng cao.

*b. Độ phân giải, tỉ lệ*

Thiết bị phần cứng dùng làm giao diện hiển thị các ảnh bản đồ là một màn hình video màu. Kích thước cửa ảnh trên màn hình bị chi phối bởi kích thước vật lý của màn hình, số hàng, cột của các điểm ảnh trên màn hình, trong bộ nhớ video và trong bộ phận trợ giúp đồ họa. Trong hầu hết các trường hợp, số lượng hàng, cột trên màn hình tương xứng với số lượng hàng, cột trong bộ nhớ video. Độ phân giải của ảnh trên màn hình là khoảng cách trên mặt đất tương ứng với một điểm ảnh. Độ phân giải này thường không liên quan đến độ phân giải của dữ liệu trong cơ sở dữ liệu. Thí dụ, nếu một lớp dữ liệu có cấu trúc raster được lưu trữ trên đĩa với số dòng cột là (4000,4000) nhưng được hiển thị trên màn hình với số dòng cột là (1000,1000) thì lúc đó ảnh hiển thị chỉ sử dụng một phần tư điểm ảnh trong mỗi hướng, do đó ảnh bị thu nhỏ kích thước lại còn một phần tư ảnh. Ngược lại, một ảnh raster trên đĩa mà có kích thước nhỏ hơn kích thước của màn hình thì khi hiển thị một số điểm ảnh có thể lặp lại để đạt được hiệu quả hiển thị và lấp đầy màn hình lớn hơn. Dữ liệu đường, điểm và các ký hiệu được liên kết với chúng được raster hóa trong suốt quá trình hiển thị.

Tỉ lệ hiển thị của một ảnh bản đồ có thể bị thay đổi do sự thu phóng của phần cứng hoặc phần mềm. Trong phần cứng, sự thu phóng đơn giản là sự phóng đại một vị trí trên màn hình mà không thay đổi độ phân giải ảnh. Những điểm ảnh đặc biệt được tái tạo và khi phóng to, ảnh trở thành hình khối. Đối với sự thu phóng trong các phần mềm thường hữu ích hơn bởi vì độ phân giải thay đổi bằng cách hiển thị lại từ cơ sở dữ liệu.

Hình ảnh xuất ra từ GIS có thể xuất đến các thiết bị khác hơn là xuất ra màn hình video, như các máy in hoặc máy vẽ. Trong một số trường hợp, các nội dung trên màn hình video được chuyển đơn giản thành một tập tin raster với cùng số lượng các điểm ảnh trên màn hình. Một số trường hợp khác, một tập tin ảnh số được tạo ra cho thiết bị phần cứng đặc biệt như máy vẽ raster hoặc vector định dạng lớn. Như một sự lựa chọn, một tập tin độc lập thiết bị được tạo ra, được biết đến như một siêu tệp đồ họa. Một định dạng thông thường cho các siêu tệp đồ họa là một tập tin PostScript. Bên cạnh việc độc lập thiết bị, các siêu tệp đồ họa duy trì độ phân giải không gian của dữ liệu, ngược với ảnh raster bị hạn chế tới độ phân giải của bộ nhớ hiển thị điều này có thể dẫn đến làm mất đi nhiều tính chất sắc xảo của ảnh. Các tập tin đồ họa độc lập thiết bị có thể khởi động được từ bất kỳ phần mềm hiển thị nào đang có sẵn.

### **Tạo trang in bản đồ**

### *Xây dựng bản đồ trong GIS*

Nguồn dữ liệu trong GIS rất phong phú và đa dạng, gồm dữ liệu trong nhiều lĩnh vực khác nhau như quản lý đất đai, nhà cửa, quản lý hành chính, quản lý tài nguyên môi trường, giáo dục, y tế... được phân thành dữ liệu không gian và dữ liệu thuộc tính có mối liên quan lẫn nhau. Để có thể xem xét một cách tổng quát mối liên hệ giữa không gian và thuộc tính, liên hệ giữa dữ liệu GIS với các đối tượng của thế giới thực cũng như việc mô tả thế giới thực một cách hiệu quả chúng ta phải chuyển nguồn dữ liệu GIS về những mô hình thu nhỏ của thế giới thực mà trong đó có thể biểu diễn cả những thông tin thuộc tính của đối tượng. Một công cụ rất hiệu quả có thể chuyển tải được tất cả các nội dung cần hiển thị đó là bản đồ.

Bản đồ được xác định như là sự mô tả đồ họa của các đối tượng trong thế giới thực đã được thay thế bằng những ký hiệu tại đúng vị trí trong không gian với 1 tỉ lệ nhỏ hơn.

Bản đồ được xem như là nguồn dữ liệu đầu vào của GIS và cũng là công đoạn cuối cùng trong quá trình làm việc với GIS, kết quả của quá trình phân tích, truy vấn, khảo sát thông tin... đều được hiển thị lên bản đồ.

Một bản đồ hoàn chỉnh là sự hợp nhất của các thành phần con có mối liên hệ mật thiết và bổ sung cho nhau. Mỗi thành phần có nhiều cách thể hiện khác nhau, vì vậy, ngoài việc chọn lựa những thành phần cần thể hiện trên bản đồ, màu sắc, cấu trúc đường nét, còn phải sắp xếp các thành phần trên một bản đồ sao cho hợp lý, khoa học, có hiệu quả khi sử dụng và nhất là phải có tính thẩm mỹ cao là điều không phải dễ dàng.

*a. Các loại bản đồ trong GIS*

Việc phân loại bản đồ tùy thuộc vào mục đích sử dụng của bản đồ, lãnh thổ và nội dung thể hiện của bản đồ.

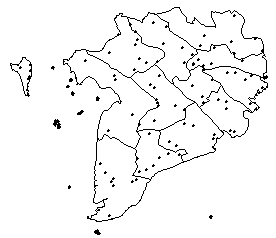
* ***Theo lãnh thổ:*** Bản đồ các châu lục, bản đồ từng quốc gia, khu vực, thấp hơn thì có bản đồ các tỉnh, quận huyện...
* ***Theo mục đích sử dụng:*** Bản đồ quân sự, bản đồ du lịch, bản đồ giáo khoa, bản đồ kinh tế vùng...
* ***Theo nội dung:*** Bản đồ địa lý chung và bản đồ chuyên đề.

Bản đồ chuyên đề chỉ thể hiện một hoặc hai chủ đề hay lớp thông tin. Để hiển thị lên trên bản đồ những đối tượng khác nhau trong không gian như đường giao thông, sông hồ, nhà cửa, rừng núi... người ta phải chọn những ký hiệu có hình dạng tương ứng để biểu thị. Thí dụ: đường giao thông trên thực tế có dạng đường nên sẽ biểu diễn lên bản đồ có dạng đường, hay vùng ao hồ, điểm độ cao....Mỗi phương pháp biểu thị khác nhau của dữ liệu sẽ tạo ra những loại bản đồ khác nhau, có thể có những bản đồ dạng điểm, dạng đường, dạng vùng hay những bản đồ hiển thị hoạt cảnh 3 chiều.

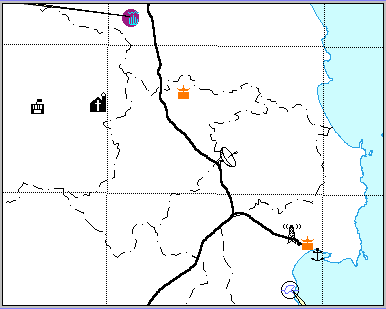
Những bản đồ thể hiện một số hoặc tất cả những loại đối tượng trong thế giới thực với mức độ quan trọng như nhau gọi là bản đồ địa lý chung.

Bản đồ địa lý chung bao gồm những thông tin về địa hình, thủy hệ, giao thông, thực vật, dân cư, ranh giới hành chính. GIS sử dụng loại là bản đồ này để tạo ra những lớp bản đồ riêng biệt và để tra cứu thông tin.

*Bản đồ điểm (Dot map)* thường sử dụng những điểm để thể hiện vị trí của các đối tượng không gian hay sử dụng mật độ các điểm bên trong vùng để thể hiện giá trị dữ liệu phân theo từng vùng. Ví dụ: bản đồ chỉ mật độ như: mật độ dân số, cây trồng, nhà cửa, bản đồ phân bố dân cư, bản đồ các trung tâm hành chính....



*Bản đồ ký hiệu hình ảnh (Picture symbol map)* sử dụng những ký hiệu tượng hình để thể hiện cho những đối tượng, thường sử dụng trong bản đồ du lịch.



*Bản đồ ký hiệu tăng dần* *(Graduated symbol map)* sử dụng kích thước của ký hiệu nhằm tạo ra thang ký hiệu tăng dần theo độ gia tăng của giá trị thuộc tính. Các giá trị của dữ liệu sẽ được thể hiện bằng độ lớn nhỏ của ký hiệu. Khi tạo thang ký hiệu cho bản đồ, cần phải chọn vùng giới hạn hiển thị, kích cở, độ rộng của ký hiệu cho hợp lý để tránh trường hợp che khuất những đối tượng khác trên bản đồ.

*Bản đồ giá trị duy nhất (Unique values map):* Bản đồ loại này sử dụng các ký hiệu khác nhau bằng màu sắc khác nhau để ký hiệu cho mỗi giá trị thuộc tính.

Bản đồ giá trị duy nhất phù hợp với các loại bản đồ sau:

* Bản đồ mô tả tên, kiểu dáng, trạng thái hoặc phân loại của một đối tượng.
* Bản đồ chứa đựng sự đo lường hoặc số lượng mà đã được phân loại.
* Bản đồ mang đặc điểm chỉ định tính duy nhất.

*Bản đồ màu tăng dần (Graduated Color map)*: Những bản đồ loại này thể hiện một loạt các ký hiệu có màu thay đổi theo tính chất liên tục và theo mức độ từ thấp đến cao nhằm tạo ra thang màu tăng dần của các giá trị thuộc tính. Kí hiệu này chỉ dành cho các thuộc tính dạng số của những đối tượng (điểm, đường, vùng). Thường sử dụng cho loại bản đồ thể hiện nhiệt độ, lượng mưa, độ cao địa hình...

*Bản đồ biểu đồ (Chart):*Các bản đồ loại này sử dụng biểu đồ (dạng tròn, dạng cột) để hiển thị những thông tin có quan hệ với nhau trong những thuộc tính khác nhau.



### *Tạo trang in bản đồ*

Một bản đồ hoàn chỉnh gồm các thành phần có mối liên hệ với nhau và bổ sung cho nhau. Vì vậy, ngoài việc chọn lựa những thành phần cần thể hiện trên bản đồ, màu sắc, dạng đường nét, còn phải sắp xếp các thành phần trên bản đồ sao cho hợp lý, khoa học, hiệu quả và thẩm mỹ. Một bản đồ có các thành phần sau:

* Nội dung: bản đồ chính, nói lên được chủ đề mà bản đồ sẽ thể hiện.
* Yếu tố hỗ trợ: bản đồ, bảng chú giải, yếu tố xuất bản.
* Cơ sở toán: khung bản đồ, lưới chiếu, tỷ lệ.
* Yếu tố bổ sung: bản đồ phụ, hình ảnh minh họa, biểu đồ,…

Trong đó, những thành phần trong nội dung và cơ sở toán là quan trọng, quyết định nội dung của bản đồ. Những thành phần trong yếu tố hỗ trợ giúp đọc được bản đồ chính, những thành phần trong yếu tố bổ sung nhằm minh họa them cho nội dung của bản đồ.

# ***CHƯƠNG 5*** LẬP TRÌNH GIS CƠ BẢN VỚI NGÔN NGỮ AVENUE

## **CƠ BẢN VỀ NGÔN NGỮ LẬP TRÌNH AVENUE**

## **Giới thiệu ngôn ngữ Avenue**

Cùng với sự ra đời của phần mềm ArcView, các nhà phát triển phần mềm ArcView đã tích hợp trong nó một ngôn ngữ lập trình Avenue đi kèm. Tuy nền tảng phát triển đều dựa trên các thành phần xây dựng sẵn bên trong ArcView, nhưng với sự hỗ trợ của ngôn ngữ lập trình Avenue, những người khai thác phần mềm ArcView không còn lệ thuộc hoàn toàn vào các công cụ có sẵn trong phần mềm ArcView nữa, Avenue cho phép đem các thành phần có sẵn này xây dựng nên các công cụ hỗ trợ khác nhau. Những người khai thác phần mềm ArcView có thể áp dụng nó trong các lĩnh vực khác nhau, ở các quốc gia khác nhau. Với các thành phần có sẵn, khi ta làm việc với ngôn ngữ lập trình Avenue trong ArcView ta không cần biết bằng cách nào người ta đã xây dựng và hiện thực nên các thành phần có sẵn trong ArcView, mà chỉ cần biết các thành phần này có những chức năng gì, hoạt động như thế nào là có thể đem các thành phần này lắp ráp lại với nhau tạo nên một ứng dụng theo ý muốn.

Ngôn ngữ lập trình Avenue dễ học và dễ sử dụng, những người có kiến thức căn bản về tin học và biết sử dụng ArcView đều có thể tiếp cận được ngôn ngữ lập trình này.

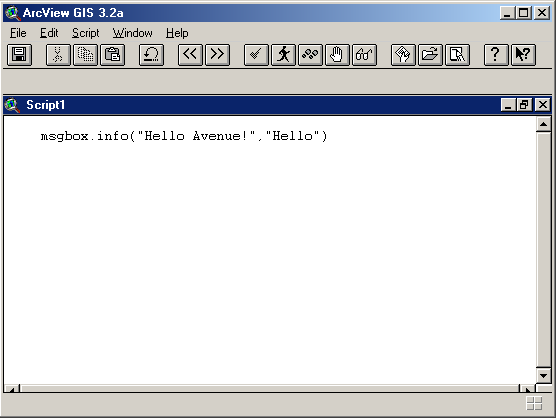
## **Các đặc điểm của ngôn ngữ Avenue**

* Ngôn ngữ lập trình Avenue là ngôn ngữ kịch bản (script) hướng đối tượng
* Ngôn ngữ lập trình Avenue không có tính nhạy cảm.
* Ngôn ngữ lập trình Avenue là ngôn ngữ biên dịch.
* Ngôn ngữ lập trình Avenue không hỗ trợ đệ quy.
* Ngôn ngữ lập trình Avenue không cho phép tạo mới hay sửa đổi các lớp có sẵn trong ArcView.
* Các kịch bản Avenue không thể thực thi bên ngoài ArcView.
* Các biểu thức trong Avenue thực thi theo chiều từ trái qua phải và không có độ ưu tiên toán tử.

## **Giao diện lập trình Avenue**

* Giao diện lập trình Avenue nằm trong Project của ArcView.
* Để mở giao diện này, ta theo các bước sau.
* Mở ArcView -> Chọn Script -> Nhấn vào nút New để tạo mới một Script. Script mới tạo có tên ngẫu nhiên là script1.
* Muốn đổi tên của Script này ta vào menu Script, trong hộp thoại này gõ vào tên script mới trong trường name.
* Trong một project ta có thể tạo ra nhiều script khác nhau.

Khởi độngArcView 🡪 Chọn *Script* 🡪 Nhấn vào nút *New* để tạo mới một Script.



Nút chạy script

Nút đặt điểm ngắt

Nút chạy từng bước

Nút xoá điểm ngắt

Nút xem script hệ thống

Nút mở một script

Nút lưu script xuống file

Hai nút trợ giúp

Các nút hỗ trợ soạn thảo

Nút lưu Project

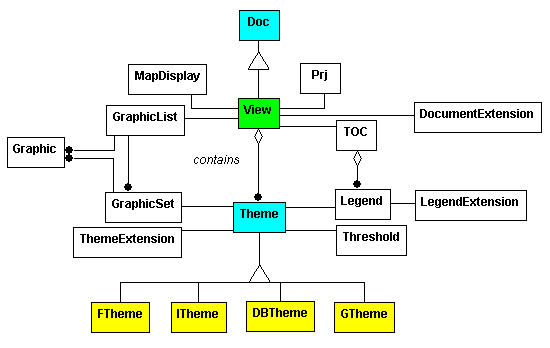
Nút dịch script

## **Lớp và đối tượng**

Avenue là ngôn ngữ kịch bản (script) hướng đối tượng. Trong ArcView, mọi thứ đều được xem là đối tượng, khi ta mở một khung nhìn (view), khung nhìn đó được gọi là đối tượng khung nhìn, khi ta thêm một theme mới vào khung nhìn, theme đó được gọi là đối tượng theme.

Trong ArcView mối quan hệ giữa các lớp được biễu diễn thông qua lược đồ mô hình đối tượng. Để có thể lập trình tốt, người lập trình phải biết cách đọc và hiểu được các lược đồ mô hình đối tượng này.

Ví dụ về một lược đồ mô hình đối tượng trong ArcView:



## **Yêu cầu và cách gọi yêu cầu trong Avenue**

Trong ArcView có hai dạng yêu cầu:

- Yêu cầu thực thể: là loại yêu cầu được gửi tới thực thể của lớp, ví dụ ta gửi một yêu cầu tới đối tượng theme để hiển thị hay che dấu theme này*. (một theme cụ thể)*

- Yêu cầu lớp: Là loại yêu cầu ta gửi trực tiếp tới lớp, ví dụ ta gửi một yêu cầu tạo mới theme tới lớp theme để tạo mới một theme. *(gửi tới lớp theme để tạo ra 1 theme)*

Trong ArcView tên của yêu cầu thường bắt đầu là một động từ, chữ cái đầu được viết hoa.

*Ví dụ:* GetProject, FindScript, AddDoc, SetValue, trong đó Get, Find, Add, Set chỉ các hành động, Project, Script, Doc, Value nhận các hành động này.

Các động từ bắt đầu của các yêu cầu dùng trong ngôn ngữ Avenue được trình bày trong bảng sau:

|  |  |
| --- | --- |
| **Động từ bắt đầu** | **Ý nghĩa** |
| Add | Thêm một đối tượng khác vào nhóm đối tượng |
| As | Chuyển đổi một đối tượng thuộc lớp này sang một đối tượng thuộc lớp khác |
| Can | Kiểm tra xem một đối tượng có thể thực hiện được một chức năng nào đó |
| Find | Tìm một đối tượng xác định và trả về tham khảo tới nó |
| Get | Trả về tham khảo tới một đối tượng |
| Has | Chỉ ra nếu một đối tượng có trạng thái xác định |
| Is | Trả về True hay False |
| Make | Tạo thực thể mới |
| Return | Tạo mới một đối tượng và trả về tham chiếu tới nó |
| Set | Gán thuộc tính vào một đối tượng |

Sau đây là cú pháp lệnh gởi yêu cầu đến một đối tượng thực hiện một hành động nào đó:

***Object.Request***

Trong đó:

* *Object* là đối tượng mà ta cần gửi yêu cầu tới,
* *Request* là yêu cầu mà ta cần gửi tới đối tượng Object.

*Ví dụ:*

Tạo mới một khung nhìn: *View.Make*

Cũng có thể kết quả trả về của một yêu cầu là một đối tượng, với đối tượng vừa trả về ta cũng có thể gửi yêu cầu tới nó.

*Ví dụ:*

*av.GetActiveDoc.GetGUI*

*Ví dụ:*

Ta cần tìm theme có tên là dgt.shp trong khung nhìn có tên là “Ban do Dong Thap” ta phải viết như sau:

*av.FindDoc(“Ban do Dong Thap”).FindTheme(“dgt.shp”)*

Viết được câu lệnh này là do ta hiểu được sơ đồ phân cấp lớp trong ArcView, muốn tìm một Theme thì ta phải biết theme đó nằm trong khung nhìn (view) nào, do nó nằm trong khung nhìn là Ban do Dong Thap nên ta tìm khung nhìn có tên “Ban do Dong Thap”, sau đó mới đi tìm theme dgt.shp trong khung nhìn này.

Trong Avenue có hàng ngàn lớp và yêu cầu khác nhau, người lập trình không cần phải nhớ hết tất cả các lớp và yêu cầu, điều quan trọng là người lập trình tìm đúng đến lớp và yêu cầu cần dùng. Trong khuôn khổ cuốn sách này chúng tôi cũng không trình bày hết tất cả các lớp và yêu cầu trong ArcView mà chỉ trình bày những lớp và yêu cầu quan trọng hay dùng, còn các lớp và các yêu cầu khác người lập trình có thể tìm kiếm dễ dàng trong phần trợ giúp của ArcView.

## **Các kiểu dữ liệu trong Avenue**

- Kiểu đối tượng số

- Kiểu đối tượng String

- Kiểu đối tượng Pattern

- Kiểu đối tượng luận lý

- Kiểu đối tượng Nil: Lớp Nil chỉ có một thực thể duy nhất thuộc về nó đó là đối tượng nil.

Trong Avenue ta hay ***dùng giá trị nil để chỉ một số yêu cầu không trả về giá trị nào cả.***

*Ví dụ:*

*t = av.FindDoc(“View1”)*

*if (t = nil) then*

*msgbox.info(“khong tim thay doi tuong khung nhin View1”, “Thong bao”)*

*end*

*- Kiểu danh sách*

Danh sách là tập hợp có thứ tự của các đối tượng chứa trong nó. Các đối tượng chứa trong sanh sách có thể ở bất kỳ dạng nào. Mỗi đối tượng trong danh sách được gọi là một phần tử, vị trí của phần tử được xác định thông qua chỉ số.

Chỉ số đầu tiên của danh sách luôn luôn là **0**. Chỉ số cuối cùng của danh sách là *List.Count -1.*

*Ví dụ:*

*alist = {1, 5, “a”, “hello world”} ‘ danh sách alist có 4 phần tử*

*blist = {6 , alist, “nice” } ‘ danh sách blist có 3 phần tử mà phần tử thứ hai là một danh sách.*

*f = alist.Get(0) ‘ lấy phần tử đầu tiên của danh sách alist, f có giá trị là 1*

*l = alist.Get(alist.Count-1) ‘ lấy phần tử cuối cùng của danh sách alist, l có giá trị là “hello world”*

Để duyệt từng phần tử trong danh sách ta hay dùng vòng lặp for

*Ví dụ:*

*For each i in alist*

*Msgbox.info(i.AsString, “Phan tu”)*

*End*

*hoặc*

*For i = 0 to alist.Count-1*

*Msgbox.info(alist.Get(i).AsString, “Phan tu”)*

*End*

Phát biểu for là một dạng phát biểu lặp trong Avenue, ta sẽ tìm hiểu phát biểu này trong phần các phát biểu của Avenue.

Ta cũng có thể thực hiện các phép ***+ (phép hợp), - (phép hiệu)*** trên danh sách

*Ví dụ:*

*alist = {1 ,2 , 4, 5}*

*blist = {3, 4, 6, 7}*

*clist = alist + blist ‘clist={1,2,4,5,3,4,6,7}*

*dlist = alist- blist ‘dlist={1 ,2 ,5}*

Ví dụ sau thực hiện việc in ra các phần tử trong một danh sách

*alist = {“pen”, “ruler”, “ink”, “table”, “chair”}*

*for each i in alist*

*msgbox.info(i, “Phan tu”)*

*end*

*- Kiểu Stack*

Giống như danh sách, Stack cũng là tập hợp các đối tượng có thứ tự nhưng Stack có điểm khác biệt là cách thức truy xuất các phần tử. Với Stack ta truy xuất các phần tử theo thứ tự Last in, First out (LILO), tức là phần tử vào sau cùng được lấy ra đầu tiên.

Lớp ***Stack*** hỗ trợ hai yêu cầu chính đó là ***push*** và ***pop***, push dùng để cho một phần tử vào Stack, pop dùng để lấy một phần tử ra khỏi Stack.

*-* *Kiểu Dictionary*

***Dictionary*** là một loại tập hợp mà ta ***truy xuất vào các phần tử của nó không theo thứ tự***, khác với danh sách dùng chỉ số là kiểu số để truy xuất vào một phần tử, Dictionary dùng ***khoá (key) có kiểu bất kỳ để xác định một phần tử trong nó***.

Khi khởi tạo một Dictionary, ta cần xác định kích thước ban đầu cho nó, trong lúc chạy ta có thể thêm vào Dictionary số lượng phần tử lớn hơn kích thước của nó, tuy nhiên điều này sẽ làm tốn thời gian để truy xuất một phần tử.

Các yêu cầu của lớp Dictionary

|  |  |  |
| --- | --- | --- |
| **Tên yêu cầu** | **Kiểu trả về** | **Ý nghĩa** |
| Make(size) | Dictionary | Khởi tạo một Dictionary có kích thước là size. |
| Add(key,Value) | Boolean | Thêm vào Dictionary một phần tử có giá trị là Value, khoá là key. Nếu khoá này đã tồn tại thì trả về false và phần tử không thêm được vào Dictionary |
| Get(key) | Object | Lấy phần tử có khoá là key trong Dictionary |
| Remove(key) |  | Xoá phần tử có khoá là key trong Dictionary |
| Count | Number | Lấy về số phần tử trong Dictionary |

*Ví dụ:*

*d = Dictionary.Make(11) ‘ tạo Dictionary có kích thước là 11d.Add("List 1", {"a", "b", "d"}) ‘ Thêm vào d một danh sách có khoá là “List 1”d.Add("List 2", {"What", "Is", "This?"}) ‘ Thêm vào d một danh sách có khoá là “List 2” d.Add("List 3", {"Goodnight", "Night", "John", "Boy"}) ‘ Thêm vào d một danh sách có khoá là “List 3”*

*aList = d.Get("List 1") ‘ aList ={“a”,”b”,”d”}**- Kiểu Bitmap*

Một bitmap là một tập hợp *có thứ tự* và *có kích thước cố định* của các *giá trị luận lý* được gọi là các *bit*. Giá trị mỗi bit cho ta biết bit đó được thiết lập hay không (nghĩa là true hay là false).

Bitmap thường được sử dụng để lựa *chọn các record trong một bảng*, với mỗi record được chọn bit tương ứng với record này được thiết lập ở giá trị true.

Bitmap còn được sử dụng để *điều khiển cách hiển thị của theme*, chẳng hạn trong lớp đường giao thông ta chỉ muốn hiển thị các con đường cao tốc, lúc đó *ta thiết lập một bitmap để chỉ chọn các con đường cao tốc trong lớp đường giao thông.*

Một số yêu cầu của lớp bitmap

|  |  |  |
| --- | --- | --- |
| **Tên yêu cầu** | **Kiểu trả về** | **Ý nghĩa** |
| Make | Bitmap | Tạo mới một bitmap |
| Count | Number | Lấy số lượng bit *được thiết lập* trong một bitmap |
| Get(i) | Boolean | Trả về true nếu bit ở vị trí i được thiết lập và ngược lại |
| Clear(i) |  | Xoá thiết lập bit ở vị trí thứ i |
| Set(i) |  | Thiết lập bit ở vị trí thứ i |
| ClearAll |  | Xoá tất cả các bit thiết lập |

*Ví dụ:*

*abitmap = BitMap.make(5) ‘ tạo ra một bitmap có 5 bit*

*abitmap.Set(3) ‘ thiết lập bit ở vị trí thứ 3 là true*

*t = abitmap.Get(3) ‘t có giá trị là true vì bit 3 được thiết lập*

## **Các câu lệnh**

## ***Câu lệnh gán***

Phát biểu gán có dạng: **biến = đối tượng**

Một biến được gán cho một đối tượng nào đó.

*Ví dụ:*

a = 5 ‘gán cho biến a giá trị 5

myView = av.FindDoc(“View1”) ‘gán đối tượng khung nhìn View1 cho biến myView.

## ***Câu lệnh điều kiện: IF … THEN…***

|  |  |
| --- | --- |
| *Dạng 1:*  ***if (biểu thức luận lý 1) then***  ***khối phát biểu***  ***end*** | *Ví dụ:*  if (i<5) then  i = i+1  end |
| *Dạng 2:*  ***if (biểu thức luận lý 1) then***  ***khối phát biểu 1***  ***else***  ***khối phát biểu 2***  ***end*** | *Ví dụ:*  if (i<5) then  i=i+1  else  i=i+2  end |
| *Dạng 3:*  ***if (biểu thức luận lý 1) then***  ***khối phát biểu 1***  ***else if (biểu thức luận lý 2) then***  ***khối phát biểu 2***  ***else***  ***khối phát biểu 3***  ***end*** | *Ví dụ:*  if (i<5) then  i = i+1  else if(i<10) then  i = i+2  else  i = i +3  end |

## ***Phát biểu lặp***

|  |  |
| --- | --- |
| **Lặp có điều kiện:**  ***while (Biểu thức luận lý)***  ***khối phát biểu***  ***end*** | *Ví dụ:*  i = 0  while (i<100)  i = i+1  end |
| **Lặp không có điều kiện**  ***for each ele in Col***  ***khối phát biểu***  ***end*** | *Ví dụ:*  for each i in 0..10 by 2  msgbox.info(i.AsString,””)  end |

Vòng lặp for duyệt qua các phần tử trong tập hợp Col, tập hợp *Col* có thể là một *khoảng nhảy, một danh sách, một bitmap, một dictionary hay một tập lớp bất kỳ*. Nếu các phần tử có thứ tự, thì vòng for duyệt theo thứ tự đó, ngược lại for duyệt theo một thứ tự bất kỳ.

*Ví dụ:*

for each i in alist

msgbox.info(i.AsString,””)

end

Avenue cung cấp hai phát biểu ***Break*** và ***Continue*** cho phép ta điều khiển bên trong vòng lặp.

## **Điều khiển giữa các script**

Avenue cung cấp hai phát biểu cho ta điều khiển giữa các script: Phát biểu Return và phát biểu Exit.

***- Phát biểu return***

Phát biểu return có dạng ***return <anObject>***. Phát biểu return trả về một đối tượng cho Script gọi. Phát biểu return luôn có đối số.

*Ví dụ:*

‘ Script gọi

re = av.run(“getAction”, “xanh”)

msgbox.info(re, “chi dan chay xe”)

‘ script getAction

if (self = “do”) then ‘đỏ

return “dung”

elseif (self =”xanh”) then ‘xanh

return “di”

else ‘vàng

return “coi chung”

end

## ***Phát biểu exit***

Phát biểu exit *kết thúc script đang chạy.*

*Ví dụ:*

theView = av.getActiveDoc

for each t in theView.GetThemes

if (t.IsVisible = false) then

exit

end

end

## **Đối tượng av và từ khoá self**

**- Đối tượng av**

Khi ta mở ArcView, ta đã tạo một đối tượng ứng dụng ArcView, đối tượng này chứa một project, trong project này chứa tất cả các đối tượng mà ta có thể làm việc với chúng.

Trong phân cấp lớp của ArcView, ***đối tượng ứng dụng*** là đối tượng ở phân lớp cao nhất, đối tượng này gọi là ***av***.

*Ví dụ:*

Lấy về đối tượng project hiện thời

theProject = av.getProject

Lấy về cửa sổ làm việc hiện hành

theView = av.getActiveDoc

av còn dùng để gọi một script trong script khác.

- ***Có 2 cách gọi script dùng av***

**+ Dùng av.run**

*Ví dụ:*

av.run(“myscript”,{}) ‘ gọi script có tên myscript cung cấp đối số là danh sách rỗng

av.run(“ascript”,{1,5,7}) ‘ gọi script có tên ascript với đối số là danh sách {1,5,7}

**+ Dùng av.delayedRun**

Để *gọi một Script sau* *một khoảng thời gian nào đó* ta dùng av.delayedRun

*Ví dụ:* script có tên là Annoy

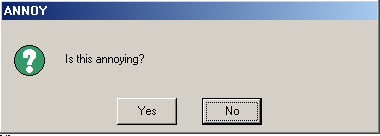
status = MsgBox.YesNo( "Is this annoying?", "ANNOY", FALSE )

if (NOT status) then

av.DelayedRun( "annoy", "", 10 ) ‘ cứ 10 giây lại gọi chính nó

end

Với script này cứ 10 giây lại hiển thị thông điệp như sau



Nếu ta kích vào nút “No” thì 10 giây tiếp theo ta lại thấy thông điệp này hiển thị lên, nếu ta kích vào nút “Yes” thì đoạn script annoy kết thúc thực thi.

**- Từ khoá self**

Self thường dùng để *tham khảo tới đối tượng gây ra sự kiện chạy script*, khi nhấn chuột vào một button thì ta có thể dùng self để tham khảo đến button này.

Self còn dùng để *lấy giá trị tham số truyền từ script khác gọi nó*.

*Ví dụ:*

‘ script1 có nội dung như sau

‘ tinh tong cua 2 so a, b

***a*** = 10

b = 20

tong = av.run(“script2”,{***a***,b})

‘ script2 có nội dung như sau

a = self.get(0) ‘lấy giá trị tham số thứ nhất của script1 (là ***a***)

b = self.get(1)

c = a + b

return c

Khi chạy script1, giá trị của tổng trả về là 30

***Ví dụ:***

***Ví dụ 1. Viết script tính tổng của các phần tử trong một danh sách gồm các đối tượng kiểu số.***

Hướng dẫn:

‘ script tinhtong

alist={1,3,5,6,3,2,25,30}

tong=0

for each i in alist

tong = tong + i

end

msgbox.info(“tong cac phan tu trong danh sach “ + tong.AsString, “Tong”)

***Ví dụ 2. Viết script tìm số lớn nhất của 3 số***

Hướng dẫn:

‘ script timmax

a=4

b=5

c=7

m=a

if ((a>=b) and (a>=c)) then

m =a

elseif ((b>=a) and (b>=c)) then

m=b

elseif ((c>=a) and (c>=b)) then

m=c

end

***Ví dụ 3. Cho một danh sách bất kỳ, tìm các phần tử có kiểu số.***

*Vd: alist={123, 45, “cd”,* “ef”, 6, “55”} 🡪 kết quả: blist = {123,45,6}

Hướng dẫn:

alist={“123”,45,”cd”,”ef”,”6”}

blist={}

for each i in alist

if (i.getClass.getClassName= “Number”) then

blist.add(i)

end

end

Yêu cầu getClass lấy về lớp của một đối tượng, yêu cầu getClassName trả về tên của một lớp.

Bài tập 1. Viết script tìm số lớn nhất của một danh sách các phần tử kiểu số.

Bài tập 2. Viết script sắp xếp tăng dần các phần tử kiểu số trong một danh sách (các phần tử trong danh sách kiểu số).

Bài tập 3. Viết script dùng Msgbox.input nhập vào n phần tử, sau đó in ra message giá trị trung bình của n phần tử này.

Bài tập 4. Viết Script đổi một chuỗi có các chữ cái đầu thành chữ hoa

Ví dụ: “truong dai hoc” sẽ được đổi thành “Truong Dai Hoc”

Bài tập 5. Viết script giải phương trình bậc 2

Bài tập 6. Viết script tính phương sai của một tập điểm.

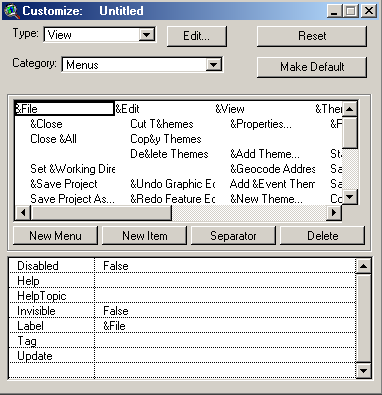
### **THAY ĐỔI GIAO DIỆN ARCVIEW**

ArcView cho phép ta thêm mới, xoá, sửa Menu, Button, Tool để phục vụ cho nhu cầu riêng. Để thêm mới, xoá, sửa giao diện, ta D-Click vào cửa sổ giao diện ArcView như hình dưới đây:



Double vào đây để thay đổi Menu, Button, Tool, Popup menu

Hộp thoại chỉnh sửa như hình sau:

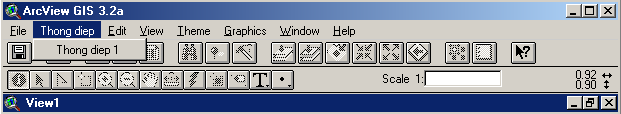


Trong hộp thoại chỉnh sửa này ta có thể thực hiện các thao tác thêm mới, chỉnh sửa, xoá các điều kiển trên giao diện.

### **Tạo Menu**

***Để tạo mới Menu*** ta làm như sau:

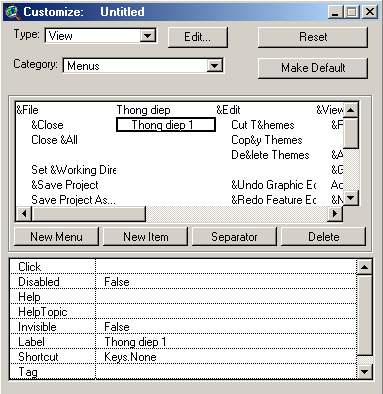
* Mở hộp thoại chỉnh sửa.
* Chọn *Menus* trong *Caterogy*
* Nhấn vào nút ***New Menu*.** Một Menu có tên ngẫu nhiên là Menu mới được tạo ra.
* Kích đôi vào mục *Label* để đổi tên của Menu, đổi tên của menu là *Thong diep*
* Để ***tạo mới một menu con*** cho menu này ta nhấn vào nút ***New Item***. Một menu con có tên là Item được tạo ra.
* Doube click vào mục Label để đổi tên của Item này, đổi tên menu là Thong *diep 1*
* Đóng hộp thoại chỉnh sửa giao diện, ta thấy xuất hiện trên menu của ArcView một menu có tên là Thong diep



Khi ta chọn vào menu con Thong diep 1 ta không thấy có sự kiện nào xảy ra, điều này do ta chưa gán một script nào cho sự kiện chọn menu con này.

Bây giờ ta tiến hành gán một script cho sự kiện chọn Menu con này.

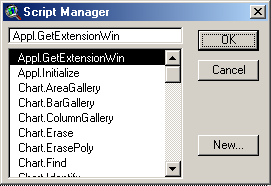
* Mở hộp thoại chỉnh sửa giao diện
* Chọn memu con Thong diep 1
* Kích đôi vào mục Click



Kích đôi vào đây để tạo tooltip

Kích đôi vào đây đặt nhãn

Kích đôi vào đây để gán Script cho sự kiện kích

* Hộp thoại yêu cầu xác định script cho menu này xuất hiện: 

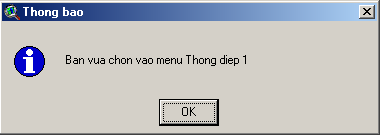
Trong hộp thoại này ta ***chọn Script cho có sẵn*** hoặc nhấn ***vào nút New để tạo mới một script cho menu này.***

* Nhấn vào nút New trong hộp thoại Script Manager để tạo mới một script. Đặt tên cho script này là Menu1
* Trong giao diện soạn thảo script, ta gõ vào script sau.

*tenMenu=self.getLabel*

*msgbox.info(“Ban vua chon vao menu ” + tenMenu , “Thong bao”)*

* Đóng giao diện soạn thảo script, quay về cửa sổ giao diện khung nhìn, ta chọn vào menu con Thong diep 1, lúc đó màn hình xuất hiện hộp thoại thông điệp như sau.



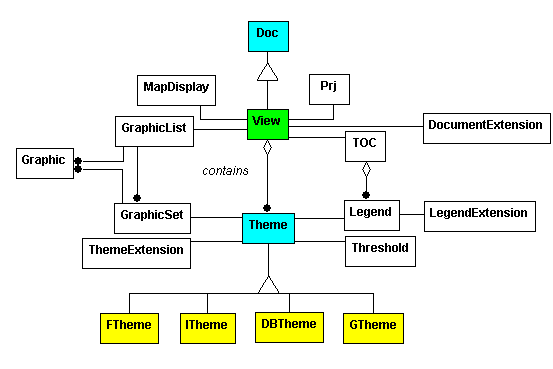
### **Xóa Menu**

***Để xoá Menu*** Thong diep ta vào hộp thoại chỉnh sửa giao diện, chọn Menu này và nhấn vào nút ***Delete.***

#### Tương tự cho Button, Tool, Popup Menu…

## **KHUNG NHÌN (VIEW) VÀ LỚP CHỦ ĐỀ (****THEME)**

## **Lược đồ mô hình đối tượng của khung nhìn (View) và lớp chủ đề (Theme)**



Nhìn vào mô hình đối tượng bên trên, ta thấy rằng View là một loại của document (DOC). Nó được chứa đựng bên trong một cửa sổ mà ta có thể mở, đóng, thay đổi kích cỡ hoặc di chuyển nó. Một View có thể chứa đựng không hoặc nhiều Theme, nó cũng có bảng nội dung (TOC) để hiển thị chú giải cho mỗi Theme, có vùng hiển thị bản đồ (Map Display) và nó cũng chứa đựng một tập các đối tượng đồ hoạ (graphics) như vòng tròn, hình chữ nhật, đường thẳng, nhãn mà ta có thể vẽ trên vùng bản đồ để làm nổi bật các đối tượng không gian trên bản đồ.

## **Tạo mới một khung nhìn**

***Để tạo mới một View*** ta sử dụng yêu cầu ***Make*** và gửi yêu cầu Make đến lớp View

***myView=View.Make***

Khi tạo ra View, ta cần đặt tên cho nó. Yêu cầu ***SetName*** cho phép ta ***đặt tên cho View***, để sử dụng yêu cầu này, cần truyền vào một đối số có kiểu String. Đối số này chính là tên View cần đặt.

***myView.SetName(“Khung nhìn bài tập chương 6”)***

Lúc này ta đã tạo được một View mới có tên là “Khung nhìn bài tập chương 6” trong đề án, bây giờ ta cần mở View này lên để có thể nhìn và thao tác trên View. Để mở View, ta cần phải lấy được cửa sổ của View.

Yêu cầu ***GetWin sẽ trả về cửa sổ của View***: ***theWindow = myView.GetWin***

Yêu cầu ***Open sẽ cho phép mở cửa sổ View***: ***theWindow.Open***

Nếu muốn cửa sổ View này ***hiển thị lên trên tất cả các cửa sổ khác của đề án***, hãy sử dụng yêu cầu Activate.

***theWindow.Activate***

## **Tìm kiếm một khung nhìn**

Tưởng tượng ta cần tìm đến một View với tên biết trước như “*Khung nhìn bài tập chương 6*”. Để làm được điều này trong ArcView, trước tiên mở cửa sổ đề án (Project) và sau đó tìm tên “*Khung nhìn bài tập chương 6* ” trong GUI View. Tương tự như vậy, để viết các lệnh tìm View trong Avenue, đầu tiên phải tìm đến đề án (Project) bằng yêu cầu GetProject

***theProject = av.GetProject***

Vì View là một hiện thực của lớp document (DOC) nên tiếp theo ta cần sử dụng yêu cầu FindDoc để tìm đến View

***myView = theProject.FindDoc(“Khung nhìn bài tập chương 6”)***

## **Gán đơn vị bản đồ và đơn vị thước đo cho khung nhìn**

Để biết được tỷ lệ của bản đồ đang hiển thị trên View, ta cần phải gán đơn vị bản đồ cho View. Sử dụng yêu cầu SetUnits cùng với đối số là đơn vị bản đồ cần gán, ví dụ như Meters , đối số này thuộc kiểu hằng số do ArcView cung cấp, ta sẽ gán đơn vị bản đồ cho View

myView = theProject.FindDoc(“Khung nhìn bài tập chương 6”)

if (myView <> nil) then

myView.SetUnits(#UNITS\_LINEAR\_METERS)

end

#UNITS\_LINEAR\_METERSlà hằng số đơn vị do ArcView cung cấp. Có thể tìm thấy danh sách các đơn vị này bằng cách mở Help của ArcView và gõ vào UnitsLinearEnum trong mục Index.

Tương tự, để gán đơn vị thước đo cho View, ta sử dụng yêu cầu .SetDistanceUnits với đối số là đơn vị cần gán.

myView.SetDistanceUnits(#UNITS\_LINEAR\_METERS)

## **Thêm lớp chủ đề (theme) vào khung nhìn**

Một khi đã có đối tượng View, ta cần phải thêm một vài lớp chủ đề vào nó để có thể hiển thị được các đối tượng không gian. Các lớp chủ đề này có thể là shapefile, Image, Acad hoặc Arc/info coverage, những loại này được nhóm lại thành hai loại data source là feature data source và image data source. Mỗi loại data source này lại được phân thành các lớp trong Avenue, những lớp đó là FTheme (chủ đề dạng feature), ITheme (chủ đề dạng Image), DBTheme (chủ đề dạng Database), và GTheme (chủ đề dạng Grid)

Khi thêm một lớp chủ đề vào một View, trước hết ta cần tìm đến View. Sử dụng yêu cầu tìm View ở trên

myView=av.GetProject.FindDoc(“Khung nhìn bài tập chương 6”)

Tiếp theo cần tạo ra Theme. Để có thể tạo ra Theme, sử dụng yêu cầu Make và gửi đến lớp SourceName để tạo ra đối tượng Sourcename. SourceName là một đối tượng của Avenue dùng để chỉ đến vị trí vật lý của tệp tin cần tạo nên Theme.

theSrcName=SrcName.Make(“C: \Ditagis\Avenue\Data\hchinh.shp”)

‘ C: \Ditagis\Avenue\Data\hchinh.shp: vị trí tệp tin cần tạo thành Theme

if (theSrcName=nil) then

Msgbox.Info(“Không tồn tại tệp tin”, “Thông báo”)

exit

end

Dùng yêu cầu Make để tạo Theme với đối số là *theSrcName*

theTheme = Theme.Make(theSrcName)

Khi đã có Theme, dùng yêu cầu AddTheme để thêm mới tạo vào View

myView.AddTheme(theTheme)

Đến lúc này, Theme mới đã được thêm vào View, tuy nhiên nó vẫn chưa hiển thị trên vùng bản đồ vì ta chưa có gán thuộc tính hiển thị cho Theme. Dùng yêu cầu SetVisible với đối số là true để gán thuộc tính hiển thị cho Theme. Nếu muốn Theme không hiển thị thì dùng yêu cầu SetVisible với đối số là false.

theTheme.SetVisible (true)

Nếu muốn cho Theme ở trạng thái kích hoạt, dùng yêu cầu SetActive với đối số là true, ngược lại thì dùng đối số là false.

theTheme.SetActive (true)

Nếu muốn cấm người dùng không được phép thay đổi hai thuộc tính hiển thị và kích hoạt, dùng yêu cầu để khoá không cho hiển thị Theme là SetVisibleLocked(true) và yêu cầu khoá không cho kích hoạt là SetActiveLocked(true).

theTheme.SetVisibleLocked(true)

theTheme.SetActiveLocked(true)

Ta có thể kết nối nhiều nguồn dữ liệu khác nhau như là một chủ đề đến khung nhìn. Sau đây là một số ví dụ:

Đối với tập tin hình ảnh

TheSrcName = SrcName.Make(“$AVDATA/ myimage.bmp”)

Đối với ArcView Shapefile

TheSrcName = SrcName.Make(“$AVDATA/ myshapefile.shp”)

Đối với CAD drawings

TheSrcName = SrcName.Make(“$AVDATA/ drawings/pacels.dwg line”)

Arc/Info Librarian layer

TheSrcName = SrcName.Make(“mylibrary.layer polygon”)

Arc/Info ArcStorm layer

TheSrcName = SrcName.Make(“mylibrary. mylibrary. layer polygon”)

Ngoài ra chúng ta còn có thể tạo đối tượng SrcName từ bảng dữ liệu chứa tọa độ x,y. Ví dụ ta có bảng toạ độ có tên coords.txt, chứa hai trường là xcoord và ycoord. Ta tiến hành tạo Theme như sau:

TheView = av.GetActiveDoc

TheVtab=av.GetProject.FindDoc(“C: \Ditagis\Dulieu\Bang\qtrac\_kk.dbf”).GetVTab

xField = theVTab.FindField(“toadoX”)

yField = theVTab.FindField(“toadoY”)

xySrc = XYName.Make(theVTab, xField, yField)

theTheme = Theme.make(xySrc)

theView.AddTheme(theTheme)

## **Tìm một lớp chủ đề trong khung nhìn**

Theme được chứa trong View, vì vậy để tìm một Theme với một tên nào đó, chẳng hạn “hchinh.shp”, trước tiên phải tìm đến View và sau đó dùng yêu cầu FindTheme với đối số thuộc kiểu String là tên của Theme cần tìm

myView = av.GetProject..FindDoc(“Khung nhìn bài tập chương 6”)

hchanhTheme = myView.FindTheme(“hchanh.shp”)

## **Thay đổi thứ tự của lớp chủ đề trong bảng nội dung (TOC) của khung nhìn**

Khi thêm các Theme vào View, thứ tự của Theme trong View sẽ ảnh hưởng đến kết quả hiển thị trên vùng bản đồ. Nếu như Theme thuộc dạng điểm (Point) nằm dưới Theme thuộc dạng vùng (Polygon), thì các đối tượng của Theme thuộc dạng điểm sẽ bị che khuất không nhìn thấy trên vùng bản đồ. Để thay đổi thứ tự của các Theme, trước hết ta phải có danh sách các Theme và dùng yêu cầu Shuffle để thay đổi vị trí của Theme

Lấy danh sách Theme trong View bằng yêu cầu GetThemes

theView = av.GetProject.FindDoc(“Khung nhìn bài tập chương 6”)

theThemeList = theView.GetThemes

Giả sử ta cần đổi chỗ của Theme đầu tiên đến vị trí cuối

themeDau=theThemeList.Get(0)

vitricuoi=theThemeList.Count

theThemeList.Shuffle(themeDau,vitricuoi)

Yêu cầu Shuffle cần hai đối số, đối số đầu là đối tượng cần thay đổi vị trí, đối số thứ hai là vị trí sẽ chuyển đối tượng đến trong danh sách.

Sau khi thực hiện lệnh này, Theme đầu tiên sẽ được dịch chuyển đến vị trí cuối trong danh sách. Tuy nhiên vị trí của Theme trong bảng nội dung của View vẫn chưa tự động dịch chuyển và bản đồ vẫn chưa thay đổi. Để hiển thị kết quả trên vùng bản đồ cần phải thực thi yêu cầu vẽ lại bản đồ (refresh) và bảng nội dung.

theView.InvalidateTOC (nil)

theDisplay=theView.GetDisplay ‘ lấy vùng hiển thị bản đồ

theView.Invalidate(true) ‘vẽ lại

Sau khi thay đổi thứ tự của các Theme, nếu muốn cấm người dùng không được phép thay đổi thứ tự vị trí của các Theme, có thể khoá các Theme bằng yêu cầu SetOrderLocked

aTOC = aView.GetTOC

aTOC.SetOrderLocked

## **Thay đổi biểu tượng của lớp chủ đề**

Biểu tượng là thành phần cơ bản của đối tượng đồ họa khi hiển thị trên khung nhìn. Ta sử dụng các biểu tượng khác nhau để phân biệt các đối tượng riêng biệt trong chủ đề. Ví dụ: sử dụng các kiểu biểu tượng dạng đường (line symbols) để thể hiện các loại đường khác nhau.

Để lấy được các biểu tượng của Theme, cần phải làm việc với Legend của Theme. Legend chứa đựng các biểu tượng mà Theme đang sử dụng. Với yêu cầu GetLegend gửi cho đối tượng Theme, sẽ lấy được Legend của Theme.

theView = av.GetActiveDoc

theTheme = theView.GetThemes.Get(0) ‘ lấy Theme đầu tiên

theLegend = theTheme.GetLegend ‘ lấy Legend của Theme

Để lấy được biểu tượng, trước tiên phải lấy danh sách các biểu tượng chứa trong Legend bằng yêu cầu GetSymbols

theListSymbol=theLegend.GetSymbols

Giả sử, lấy biểu tượng đầu tiên và thay đổi màu sắc của nó thành màu xanh

theSymbol=theListSymbol.Get(0)

theBlueColor=color.GetBlue ‘ lấy màu xanh

theSymbol.SetColor(theBlueColor) ‘ gán màu symbol thành màu xanh

Để bản đồ thể hiện được sự thay đổi của biểu tượng, cần tác động lên sự thay đổi của Theme bằng yêu cầu UpdateLegend

theTheme.UpdateLegend

## **Xoá các lớp chủ đề khỏi khung nhìn**

Nếu không muốn một Theme nào đó tiếp tục tồn tại trong View, có thể xoá nó bằng yêu cầu DeleteTheme với đối số là Theme cần xoá và gửi đến đối tượng View

myView = av.GetProject.FindDoc(“Khung nhìn bài tập chương 6”)

theTheme = myView.FindTheme(“hchanh.shp”)

Tuy nhiên việc xoá Theme là việc cần phải cân nhắc, vì vậy trước khi xoá cần phải xác nhận có thực sự xoá không, nếu đồng ý thì lúc đó mới xoá, nếu không đồng ý thì không xoá.

xacnhan = Msgbox.YesNo(“ thực sự muốn xoá Theme ” + theTheme + “?”, “Xoá theme”, true)

if (xacnhan=true) then

myView.DeleteTheme(theTheme)

end

Bây giờ nếu muốn xoá các Theme đang ở trạng thái kích hoạt, đầu tiên phải gửi yêu cầu GetActiveThemes đến đối tượng View chứa các Theme đó để lấy danh sách các Theme cần xoá.

myView = av.GetProject.FindDoc(“Khung nhìn bài tập chương 6”)

listTheme=myView.GetActiveThemes

Sau đó dùng vòng lặp đi hết danh sách Theme này để lần lượt xoá các Theme

For each theTheme in listTheme

xacnhan = Msgbox.YesNo(“ thực sự muốn xoá Theme ” + theTheme + “?”, “Xoá theme”, true)

if (xacnhan=true) then

myView.DeleteTheme(theTheme)

end

end

Tương tự, nếu muốn xoá hết các Theme trong View thì dùng yêu cầu GetThemes để lấy danh sách Theme và sau đó dùng vòng lặp duyệt qua danh sách Theme và xoá nó ra khỏi View.

## **Chọn đối tượng không gian của lớp chủ đề bằng chuột**

Nếu như đã từng sử dụng qua công cụ Select Feature trên thanh Toolbar của View trong ArcView, chắc hẳn sẽ thấy công cụ trên cho phép ta chọn các đối tượng không gian của các Theme đang kích hoạt bằng chuột với 2 cách: chọn bằng điểm và chọn bằng hình chữ nhật.

Để có thể chọn được các đối tượng không gian của các Theme đang kích hoạt, trước hết phải lấy danh sách các Theme đang được kích hoạt trong View, tiếp theo lấy về điểm chọn hoặc vùng chọn và sau đó dùng điểm chọn hoặc vùng chọn này để chọn các đối tượng trên các Theme đang kích hoạt.

Lấy danh sách các Theme đang kích hoạt

myView = av.GetProject.FindDoc(“Khung nhìn bài tập chương 6”)

listActiveThemes=myView.GetActiveThemes

Lấy vùng chọn bằng cách gửi yêu cầu ReturnUserRect đến View

theRect = myView.GetDisplay.ReturnUserRect

Chọn các đối tượng trên các Theme đang kích hoạt

For each theTheme in listActiveThemes

theTheme.SelectByRect(theRect, #VTAB\_SELTYPE\_NEW)

end

*‘* #VTAB\_SELTYPE\_NEW: hằng số kiểu chọn đối tượng do ArcView cung cấp

Bảng hằng số các kiểu chọn:

|  |  |
| --- | --- |
| **Kiểu chọn** | **Ý nghĩa** |
| #VTAB\_SELTYPE\_NEW | Chọn các đối tượng thoả điều kiện trong tập toàn bộ đối tượng của Theme |
| #VTAB\_SELTYPE\_AND | chọn các đối tượng từ tập đã được chọn của Theme |
| #VTAB\_SELTYPE\_OR | Chọn các đối tượng thoả điều kiện trong tập toàn bộ đối tượng của Theme, sau đó thêm tập này vào tập đã chọn trước đó của Theme |
| #VTAB\_SELTYPE\_XOR | Chọn các đối tượng thoả điều kiện trong tập toàn bộ đối tượng của Theme, sau đó XOR tập này với tập đã chọn trước đó của Theme |

Bên trên là đoạn lệnh chọn đối tượng bằng cách dùng hình chữ nhật. Để có thể vừa dùng điểm vừa dùng hình chữ nhật như công cụ Select Feature của ArcView, thêm yêu cầu ReturnUserPoint vào đoạn lệnh trên nếu như kiểm tra thấy hình chữ nhật trả về bằng Null

myView = av.GetProject.FindDoc(“Khung nhìn bài tập chương 6”)

theRect = myView.ReturnUserRect

if ( theRect.IsNull) then

thePoint=myView.GetDisplay.ReturnUserPoint

listActiveThemes=myView.GetActiveThemes

For each theTheme in listActiveThemes

theTheme.SelectByPoint(thePoint, #VTAB\_SELTYPE\_NEW)

end

else

For each theTheme in listActiveThemes

theTheme.SelectByRect(theRect, #VTAB\_SELTYPE\_NEW)

end

end

Để chạy được Script này, cần phải gán nó vào sự kiện Apply của công cụ Tool trong cửa sổ View.

## **Chọn đối tượng không gian của lớp chủ đề bằng các hình đồ họa**

Có một cách khác để chọn các đối tượng trên Theme là sử dụng hình đồ hoạ như vòng tròn, đa giác hoặc đường thẳng. Ví dụ muốn chọn các trường học trong phạm vi có bán kính 10 km từ trung tâm của thị trấn. Đầu tiên phải vẽ ra một vòng tròn có tâm là trung tâm của thị trấn và có bán kính 10km. Tiếp theo dùng vòng tròn này để chọn các trường học trong lớp chủ đề trường học.

Giả sử ta đã vẽ vòng tròn trên View bằng công cụ vẽ vòng tròn của ArcView và nó đang ở trạng thái kích hoạt (được chọn). Về cách tạo ra graphics, sẽ được học trong chương kế. Bây giờ muốn dùng vòng tròn này để chọn các đối tượng của lớp trường học. Để làm việc này, trước tiên phải có được View hiện hành và lớp chủ đề trường học

myView = av.GetProject.FindDoc(“Khung nhìn bài tập chương 6”)

truongTheme=myView.FindTheme(“truong.shp”)

Tiếp theo, cần phải có được vòng tròn. Vì vòng tròn thuộc lớp Shape và được quản lý trong danh sách Graphics của View nên cần phải có danh sách Graphics của View để có thể lấy được vòng tròn. Gửi yêu cầu GetGraphics đến View, sẽ nhận lại danh sách các Graphics

listGraphics = myView.GetGraphics

và với yêu cầu GetSelected, sẽ có được danh sách các graphics đang ở trạng thái kích hoạt.

listSelGrp = listGraphics.GetSelected

để lấy được vòng tròn, gửi yêu cầu Get(0).GetShape đến listSelGrp

theCircle = listSelGrp.Get(0).GetShape

Lúc này ta có thể chọn các đối tượng trên Theme nằm trong phạm vi của vòng tròn bằng cách gửi yêu cầu SelectByShapes đến Theme trường học với 2 đối số là danh sách các Shapes và kiểu chọn. Để tạo ra danh sách chứa các shapes (trong trường hợp này có một Shape là theCircle), cần khởi tạo danh sách và thêm theCircle vào nó.

listShapes={}

listShapes.Add(theCircle)

Gửi yêu cầu SelectByShapes đến Theme trường học

TruongTheme.SelectByShapes(listShapes, #VTAB\_SELTYPE\_NEW)

## **Chọn đối tượng không gian lớp chủ đề bằng một lớp chủ đề khác**

Giả sử có vài trung tâm thị trấn, để tìm các trường học cách các trung tâm thị trấn trên một bán kính 10km. Thay vì phải vẽ từng vòng tròn cho các thị trấn rồi dùng các vòng tròn này để chọn các trường học, vẫn có một cách khác để thực hiện điều này, bằng cách sử dụng phương pháp chọn đối tượng trên một Theme nhờ vào các đối tượng của một Theme khác qua yêu cầu SelectByTheme.

Giả sử đã có hai Theme tồn tại trên View là Theme trung tâm thị trấn (thitran.shp) và Theme trường học (truong.shp)

myView = av.GetProject.FindDoc(“Khung nhìn bài tập chương 6”)

thitranTheme = myView.FindTheme(“thitran.shp”)

truongTheme = myView.FindTheme(“truong.shp”)

gửi yêu cầu SelectByTheme đến Theme trường học với 4 đối số là theme thị trấn, kiểu quan hệ không gian, khoảng cách quan hệ và kiểu chọn.

truongTheme.SelectByTheme(thitranTheme, #FTAB\_RELTYPE\_ISWITHINDISTANCEOF, 10, #VTAB\_SELTYPE\_NEW)

Các hằng số kiểu quan hệ không gian giữa 2 Theme:

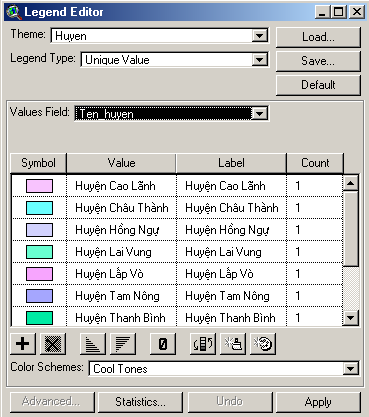
|  |  |
| --- | --- |
| Kiểu quan hệ | Ý nghĩa |
| #FTAB\_RELTYPE\_INTERSECTS | Chọn các đối tượng trên Theme cần chọn sao cho các đối tượng này giao nhau với các đối tượng trên Theme chọn |
| #FTAB\_RELTYPE\_COMPLETELYCONTAINS | Chọn các đối tượng trên Theme cần chọn sao cho các đối tượng này chứa một hoặc nhiều đối tượng đã được chọn trên Theme chọn |
| #FTAB\_RELTYPE\_CONTAINSTHECENTEROF | Chọn các đối tượng trên Theme cần chọn sao cho các đối tượng này chứa trung điểm của một hoặc nhiều đối tượng đã được chọn trên Theme chọn |
| #FTAB\_RELTYPE\_ISCOMPLETELYWITHIN | Chọn các đối tượng trên Theme cần chọn sao cho các đối tượng này nằm hoàn toàn trong một hoặc nhiều đối tượng đã được chọn trên Theme chọn |
| #FTAB\_RELTYPE\_HASCENTERWITHIN | Chọn các đối tượng trên Theme cần chọn sao cho các đối tượng này có trung điểm nằm trong các đối tượng đã được chọn trên Theme chọn |
| #FTAB\_RELTYPE\_ISWITHINDISTANCEOF | Chọn các đối tượng trên Theme cần chọn sao cho các đối tượng này nằm trong phạm vi bán kính tính từ các đối tượng đã được chọn trên Theme chọn một khoảng cho trước. |

Các yêu cầu thường được sử dụng với View

|  |  |  |
| --- | --- | --- |
| Yêu cầu | Kết quả | Ý nghĩa |
| View.Make | View | Tạo mới một đối tượng View |
| aView.AddTheme (aTheme) |  | Thêm mới một Theme vào View |
| aView.DeleteTheme (aTheme) |  | Xoá Theme khỏi View |
| aView.FindTheme (aName) | Theme | Tìm một Theme trong View theo tên của Theme |
| aView.GetActiveThemes | List | Trả về danh sách các Theme đang ở trạng thái kích hoạt |
| aView.GetDisplay | GetDisplay | Trả về Display của View |
| aView.GetGraphics | List | Trả về danh sách các graphics đang có trên View |
| aView.GetThemes | List | Trả về danh sách các Theme trong View |
| aView.GetUnits | UnitsLinearEnum | Trả về đơn vị bản đồ đang được sử dụng trong View |
| aView.ReturnScale | Number | Trả về tỷ lệ của bản đồ |
| aView.ReturnUserCircle | Circle | Trả về vòng tròn do người dùng vẽ lên trên View, yêu cầu này phải được sử dụng với một Tool |
| aView.ReturnUserLine | Line | Trả về đường thẳng do người dùng vẽ trên View |
| aView.ReturnUserPolygon | Polygon | Trả về vùng khép kín do người dùng vẽ trên View |
| aView.ReturnUserPolyLine | PolyLine | Trả về các đường thẳng do người dùng vẽ trên View |
| aView.ReturnUserRect | Rect | Trả về hình chữ nhật do người dùng vẽ trên View |
| aView.Select |  | Cho phép người dùng kích hoạt một đối tượng graphic trên View |

***Ví dụ:***

***Ví dụ 1: Thực hiện thay đổi Legend cho lớp Huyen***

Script được viết như sau

Loại chú giải

Scheme

Tên trường

Tên chủ đề

theView = av.GetActiveDoc

theTheme = theView.FindTheme(“huyen”) ‘ lấy về theme

theLegend = theTheme.GetLegend ‘ lấy về Legend hiện thời của huyện

‘ thiết lập kiểu Legend là Unique Value

theLegend.SetLegendType(#LEGEND\_TYPE\_UNIQUE)

theLegend.Unique(theTheme,"Ten\_huyen")

' Bỏ đi những giá trị Null nếu có

theLegend.SetNullValue("Ten\_huyen","")

' Lấy danh sách Scheme màu sắc

theColorSchemes = SymbolList.GetPreDefined(#SYMLIST\_TYPE\_COLORSCHEME)

‘ tìm Scheme màu có tên là “Cool Tones”

for each scheme in theColorSchemes

if (scheme.GetName = "Cool Tones") then

theColorScheme = scheme

end

end

‘ thiết lập Scheme màu cho Legend

theLegend.GetSymbols.RandomSavedSymbols(theColorScheme)

' Cập nhật lại bảng TOC

theTheme.UpdateLegend

***Ví dụ 2: Ví dụ sau thiết lập Font cho TOC***

‘ tạo bộ quản lý danh sách các Font

theFM = FontManager.The

‘ Lấy về danh sách các Font

theFontlist = theFM.ReturnFamilies

‘ Dùng một hộp danh sách liệt kê các Font để cho người sử dụng có thể chọn

theFont = msgbox.ListAsString (theFontlist,"Select a Font", "Set TOC Font")

if (theFont = nil) then ‘ trường hợp không chọn Font nào

exit

end

‘ lấy về danh sách các kiểu của Font

theStyles = theFM.ReturnStyles (theFont)

‘ Dùng một hộp danh sách liệt kê các kiểu của Font cho người dùng chọn

theStyle = msgbox.ListAsString (theStyles,"Select a Font Style","Set TOC Font")

if (theStyle = nil) then exit end ‘ không chọn kiểu nào

‘ chọn kích thước Font

theSize = msgbox.input("Enter a Font Size","Set TOC Font","12")

if (theSize = nil) then exit end ‘ trường hợp không chọn kích thước

‘ thiết lập Font cho TOC

TOCFont = TextSymbol.Make

TOCFont.SetFont (NFont.Make(TheFont.AsString,TheStyle.AsString))

TOCFont.SetSize( theSize.AsNumber)

TOC.SetDefaultSymbol(TOCFont)

‘ Cập nhật lại các View đang mở

allDocs = av.GetProject.GetDocs

for each v in allDocs

if (v.is(View) and (v.GetWin.IsOpen)) then

'only open views already open or iconized

v.GetWin.Close

v.GetWin.Open

end

end

***Bài tập***

Bài tập 1. Viết script tạo một View mới có tên là DongThap1, trong View này thêm vào các theme lớp huyen, truong và lớp dgt

Bài tập 2.Với các trường học trong View DongThap1, đổi biểu tượng cho theme truonghoc.shp sang màu đỏ.

Bài tập 3. Viết Script cho biết trong thị xã Sa Đéc có bao nhiêu trường học.

Bài tập 4.Viết Script hiển thị các trường học nằm cách quốc lộ 1A là 500 m.

**TÀI LIỆU THAM KHẢO**

1. Trần Vĩnh Phước, *GIS đại cương - phần lý thuyết*, NXB Đại học Quốc gia Tp. HCM, 2003
2. Trần Vĩnh Phước, *GIS đại cương - phần thực hành*, NXB Đại học Quốc gia Tp. HCM, 2005.
3. Lập trình Avenue, Đại học Bách Khoa TP.HCM.
4. Đặng Văn Đức, *Hệ thống thông tin địa lý, NXB. Khoa học và Kỹ thuật*, Hà Nội, 2001.
5. Ngô Đạt Tam: “Bản đồ học”, Nhà xuất bản Giáo dục 1986.
6. Trần Tấn Lộc, Lê Tiến Thuần, Bản đồ học chuyên đề, NXB Đại học Quốc gia TP. HCM, 2004.
7. Nguyễn Thế Thận, Cơ sở hệ thống thông tin địa lý, NXB Khoa học & Kỹ thuật, Hà Nội, 2002.
8. Ngô Đạt Tam: “Bản đồ học”, Nhà xuất bản Giáo dục 1986.
9. Keith C. Clarke: Getting Started with Geographic Information Systems, Prentice Hall 1999.