

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
Одеський державний екологічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до самостійної роботи студентів
при виконанні лабораторних занять
з дисципліни " Геоінформатика і ГІС "

Одеса, 2016

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
Одеський державний екологічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до самостійної роботи студентів
при виконанні лабораторних занять
з дисципліни " Геоінформатика і ГІС "

Затверджено
на засіданні кафедри інформаційних
технологій
Протокол № 11

від "12" травня 2016 р.

Зав. кафедрою:
_____ Препеліца Г.П.

Затверджено
на засіданні методичної комісії
факультету комп'ютерних наук
Протокол № __

від " __ " _____ 2016 р.

Декан:
_____ Коваленко Л.Б.

Одеса, 2016

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
Одеський державний екологічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до самостійної роботи студентів
при виконанні лабораторних занять
з дисципліни " Геоінформатика і ГІС "

Затверджено
на засіданні методичної комісії
факультету комп'ютерних наук

Методичні вказівки до самостійної роботи студентів з дисципліни " Геоінформатика та ГІС " для студентів заочної форм навчання за спеціальністю – Інформаційні управляючі системи та технології.
/ С. М. Онищенко – Одеса, ОДЕКУ, 2016. – 76 с.

Зміст

1. Загальна частина	
1.1 Передмова	5
1.2 Зміст дисциплін	5
1.3 Перелік навчальної літератури	6
1.4 Перелік знань та вмінь	7
2. Організація самостійної роботи студента	
2.1 Рекомендації студенту по роботі над дисципліною " Геоінформатика та ГІС "	8
2.2 Загальні поради	8
2.3 Зміст звіту по роботі	8
Завдання №1 «Точкові об'єкти геоінформаційної бази»	9
Завдання №2 «Характеристики геоінформаційних об'єктів»	32
Завдання №3 «Тематичні карти»	65
3. Організація контролю знань та вмінь студентів	
3.1. Система контролю знань та вмінь студентів	75
3.2 Форми контролю знань та вмінь студентів	75

1. ЗАГАЛЬНА ЧАСТИНА

1.1 Передмова

Методичні вказівки призначені для студентів заочної форми навчання. Мета виконання самостійних робіт – закріплення теоретичного лекційного матеріалу та придбання практичних навичок розроблення геоінформаційних систем.

Дисципліна «Геоінформатика та ГІС» є вибірковою дисципліною у напрямку бакалаврської підготовки «Комп'ютерні науки» та належить до професійного та практичного циклу підготовки.

Методичні вказівки містять рекомендації по вивченню розділів дисципліни, контрольні запитання та завдання. Окремі завдання контрольної роботи підкріплені прикладами розв'язання на ПЕОМ.

Загальний обсяг навчального процесу в годинах, рівень знань та вмінь при вивченні дисципліни визначаються освітньо-професійними програмами.

Дисципліна базується на знаннях та вміннях, що були набуті студентами при вивченні наступних навчальних дисциплін: «Основи програмування та алгоритмічні мови», «Гіодезія та картографування», «Цифрова картографія».

1.2 Зміст дисциплін

Тема.1 СТАНОВЛЕННЯ І ЕТАПИ РОЗВИТКУ ГЕОІНФОРМАТИКИ

Поняття об геоінформатике і її зв'язок з іншими науками, технологіями і виробництвом. Історія ГІС за рубежем. Військові застосування. Нерегулярна триангуляційна мережа. Виробники комерційних ГІС. [1, стор. 9-21], [2, стор. 21-35], [3, стор. 8-17]

Тема.2 СКЛАДОВІ ЧАСТИНИ ТА ФУНКЦІЇ ГІС

Структура і функції ГІС. Системи автоматизованого проектування. Автоматизовані довідково-інформаційні системи. [1, стор. 42-43], [4, стор. 364-366]

Тема.3 ВИДИ ДАНИХ В ГІС

Загальні підходи до представлення просторових об'єктів в ГІС. Координатні дані. Атрибутивні дані. [4, стор. 136-141]

Тема.4 МОДЕЛІ ДАНИХ

Растрова модель даних. Векторна модель даних. Мережеві моделі. Відмінність між векторною і растровою моделями. [1, стор. 62-83], [3, стор. 116-133]

Тема.5 МОДЕЛЮВАННЯ ПОВЕРХОНЬ

Псевдотривимірні моделі. Тривимірні моделі. [4, стор. 163-167]

Тема.6 ОСНОВНІ ЕТАПИ СТВОРЕННЯ ГІС

Процес введення даних в ГІС. Особливості оцифрування геологічної карти. Перевірка помилок. Побудова топології. Генералізація. [4, стор. 198-212], [3, стор. 190-194]

Тема.7 ОБЛАСТІ ЗАСТОСУВАННЯ ГІС

ГІС і охорона навколишнього середовища. Використання ГІС-технологій у медицині. ГІС для міських служб. ГІС у геодезії та картографії. ГІС для військових цілей і розвідки. ГІС залізничного транспорту. [5, стор. 10-28]

Тема.8 СУПУТНИКОВІ ТЕХНОЛОГІЇ

Супутникова радіонавігаційна система NAVSTAR-GPS. Супутникова радіонавігаційна система «ГЛОНАСС». [6, стор. 111-193]

Тема.9 ЗАСОБИ РОЗРОБКИ ГІС

Класифікація програмного забезпечення ГІС. Настільна картографічна система MapInfo. [5, стор. 38-45]

1.3 Перелік навчальної літератури

При вивченні даного курсу використовується наступна **навчальна та методична література:**

Основна

1. Основы геоинформатики: В 2 кн. Кн. 1: Учебное пособие для студентов вузов / Е.Г. Капралов, А.В. Кошкарёв, В.С. Тикунов и др.; Под ред. В.С. Тикунова. — М.: Издательский центр «Академия», 2004. — 352 с. цв. ил: ил.
2. Королев Ю.К. Общая геоинформатика. Часть I. Теоретическая геоинформатика. Выпуск 1. — М.: СП ООО Дата+, 1998. — 118 с.
3. ДеМерс, Майкл Н. Географические Информационные Системы. Основы.: Пер. с англ. — М.: Дата+, 1999. — 490 с.
4. Геоинформатика / Иванников А.Д., Кулагин В.П., Тихонов А.Н., Цветков В.Я. — М.: МАКС Пресс, 2001. — 349 с.
5. Замай С.С., Якубайлик О.Э. Программнообеспечение и технологии геоинформационных систем: Учебное пособие. — Красноярск: Краснояр. гос. ун-т, 1998. — 110 с.
6. Яценков В.С. Основы спутниковой навигации. Системы GPS NAVSTAR и ГЛОНАСС. — М: Горячая линия-Телеком, 2005. — 272 с: ил.
7. Лайкин В.И., Упоров Г.А. Геоинформатика: учебное пособие / Лайкин В.И., Упоров Г.А. — Комсомольск-на-Амуре: Изд-во АмГПУ, 2010. — 162 с.

Допоміжна

8. Светличный А.А., Андерсон В.П., Плотницкий С.В., Географические информационные системы: технология и приложения. - Одесса: Астропринт, 1997.- 196с.
9. Кошкарёв А.В., Тикунов В.С. Геоинформатика. М.: «Картгеоцентр», 1993.
10. Цветков В.Я. Геоинформационные системы и технологии. М.: «Финансы и статистика», 1998.
11. Информационный бюллетень ГИС-ассоциации.
12. Берлянт А.М. Геоинформатика. М.: «Картгеоцентр», 1996.
13. Изучение ГИС. Методология ARC/INFO, 1995.
14. Основы ГИС. Теория и практика. М.: WinGis, 1995.

1.4 Перелік знань та вмінь

Мета дисципліни - ознайомлення студентів з основними принципами, методами та можливостями геоінформаційних технологій, існуючими прикладами і засобами, що реалізують ГІС технології, привиття їм практичних навичок користування цими засобами при проведенні геоінформаційних досліджень.

В останнє десятиріччя технологія географічних інформаційних систем (ГІС технологія) отримала широке розповсюдження в багатьох країнах світу. Спектр застосування цих технологій дуже широкий.

В результаті вивчення дисципліни студент **повинен знати**: предмет досліджень геоінформатики, зв'язок з іншими дисциплінами; можливості застосування ГІС технологій; можливості сучасних ГІС пакетів; представлення атрибутивної та просторової інформації в ГІС; можливості тематичного картографування в ГІС; основні напрямки розширення аналітичних можливостей ГІС.

В результаті вивчення дисципліни студент **повинен вміти**: вводити та редагувати просторові дані; вводити та редагувати табличні (атрибутивні) дані; проводити сумісний аналіз просторових та атрибутивних даних; створювати тематичні карти.

2. ОРГАНІЗАЦІЯ САМОСТІЙНОЇ РОБОТИ СТУДЕНТА

2.1 Рекомендації студенту по роботі над дисципліною " Геоінформатика та ГІС "

Основною формою навчання студента є самостійна робота (СРС) над навчальним матеріалом. СРС повинна сприяти активізації творчого мислення студентів, формуванню високої культури розумової праці, підвищенню самостійності студентів та індивідуалізації процесу навчання. СРС складається з наступних елементів: вивчення матеріалу по підручниках, конспектах лекцій, виконання лабораторних та контрольних робіт. Студент може звертатися до викладача з питаннями для одержання письмової чи усної консультації. Однак студент повинен пам'ятати, що тільки при систематичній і завнятій самостійній роботі допомога викладача виявиться досить ефективною.

2.2 Загальні поради

– зміст кожної теми курсу вивчається за допомогою наведеного у розділі переліку навчальної літератури, який складається з основної та допоміжної літератури.

– завдання з контрольної роботи виконувати згідно з вимог, що наведені на початку кожного завдання.

– якщо виникли питання при вивченні теоретичного матеріалу або при виконанні контрольної роботи, можна звернутися до викладача, який читав установчі лекції.

2.3 Зміст звіту по роботі

Звіт виконаної самостійної роботи повинен містити розв'язок задач практичної частини з повним описом їх виконання та скриншотами результатів.

Завдання № 1

Тема: Точкові об'єкти геоінформаційної бази

Мета роботи

В геоінформаційних базах даних крім атрибутивних даних, які зберігаються як правило у достатньо відомих форматах СУБД, зберігаються також і географічні дані. Різні ГІС системи зберігають ці дані застосовуючи різні методи та підходи. Одні використовують звичайні формати СУБД (наприклад, AtlasGIS - DBase), дозволяючи користувачу безпосередньо звертатись до цих даних та обробляти їх, інші використовують свої формати збереження географічних даних, надаючи користувачеві можливості їх обробки лише засобами системи. Але всі геосистеми оснащені засобами обробки не тільки атрибутивних, а й географічних даних. Зазвичай побудова геоінформаційних баз даних є пошаровою. Географічні об'єкти розподіляються пошарово залежно від типу та необхідного набору атрибутивних даних. Найбільш поширені типи об'єктів географічних даних, що застосовуються в геоінформаційних системах – це точка, лінія (ламана), полігон. Деякі системи застосовують й інші типи об'єктів, наприклад, дуга, центроїд, мітка та інше.

Метою даної роботи є ознайомлення студентів з можливостями та особливостями роботи з географічними даними типу “точка”.

Завдання до самостійної роботи

Завдання – надати необхідні знання та виробити навички у студентів щодо роботи з географічними об'єктами геоінформаційних баз на прикладі об'єктів типу “точка”.

Після виконання завдання студенти повинні:

- *знати* можливі шляхи побудови геоінформаційних шарів бази даних, джерела помилок, що впливають на точність занесення географічних об'єктів, алгоритми перерахунку звичайних географічних координат в десяткові та навпаки;

- *вміти* будувати шари географічних даних типу “точка” та заносити відповідну атрибутивну інформацію різними способами,

здійснювати перерахунок координат.

Робота спирається на знання й уміння, отримані при вивченні "Основ СУБД".

За завданням студенту необхідно знайти географічні координати переліку деяких точкових об'єктів. Причому, потрібно знати не тільки широту та довготу (градуси, хвилини, секунди) об'єктів, а й проекцію в якій ці координати отримані. Як джерела інформації можуть використовуватись географічні карти, атласи, геоінформаційні системи, тощо.

Методичні вказівки

Способи занесення географічної інформації. Географічні дані можуть бути занесені до геоінформаційної бази різними способами, в залежності від можливостей самої ГІС. Це, наприклад:

- безпосереднє ручне нанесення географічних об'єктів на шар карти за географічними координатами засобами геоінформаційної системи;
- створення точкових об'єктів засобами системи за наперед занесеними до атрибутивної бази координатами;
- цифрування за растровою "підложкою";
- імпорт даних з других геоінформаційних систем;
- геокодування.

Перелічені способи є найбільш поширеними серед більшості ГІС. Крім того ГІС можуть застосовувати такі засоби як дигітайзери, електронні теодоліти, пристрої GPS та інше. До того ж, ГІС, що мають вбудовані мови програмування дозволяють значно прискорити роботи з занесення географічної інформації, автоматизуючи цей процес за допомогою розроблених додатків.

Деякі з перелічених способів ми розглянемо в подальшому, а в цій роботі зупинимось на об'єктах типу "точка", та на найбільш поширених способах занесення цих об'єктів до геоінформаційної бази:

- безпосереднє ручне нанесення географічних об'єктів на шар карти за географічними координатами засобами геоінформаційної системи;
- створення точкових об'єктів засобами системи за наперед занесеними до атрибутивної бази координатами.

Причому будемо робити це як в ручному режимі, так і програмно. Таким чином робота буде складатись з двох частин: ручної та програмної побудови шару точкових об'єктів з заповненням їх атрибутів.

2.1 Ручна побудова шару точкових об'єктів

В будь якому разі незалежно від способу занесення даних необхідно створити шар для занесення об'єктів та визначити перелік їх атрибутивних даних. Студент має право заносити будь-яку атрибутивну інформацію

стосовно завданих об'єктів, але п'ять видів атрибутивної інформації мають бути присутні в шарі обов'язково. Це:

1. найменування об'єкту;
2. широта об'єкту географічна;
3. широта об'єкту десяткова;
4. довгота об'єкту географічна;
5. довгота об'єкту десяткова.

Необхідно зауважити, що деякі геоінформаційні системи при роботі з географічними об'єктами оперують не з географічними, а з десятковими координатами. Формула перерахунку географічних координат в десяткові така:

$$dd = d + (m/60) + (s/3600), \quad (2.1)$$

де dd – десяткові градуси, d , m , s – географічні відповідно градуси, хвилини, секунди.

Зворотній перерахунок здійснюється за наступним алгоритмом:

$$\begin{aligned} d &= dd \setminus 1 \\ dd &= 60 * (dd - d) \\ m &= dd \setminus 1 \\ s &= 60 * (dd - m) \end{aligned}$$

де позначення згідно (1), “ $\setminus 1$ ” – операція визначення цілої частини значення.

Для того щоб створити шар об'єктів засобами системи вибираємо підпункт “Нова таблиця” меню “Файл”. Створення нової таблиці або шару об'єктів відбувається в три кроки. На кожному кроці необхідно заповнити елементи відповідного діалогового вікна. Відповідні вікна представлені на рисунках 1-3.



Рис.1. Діалогове вікно першого шагу створення нової таблиці

Перше вікно дозволяє визначити форму подання шару чи таблиці для занесення даних після закінчення процедури створення. Шар може бути наданий в вигляді таблиці для заповнення атрибутів об'єктів, чи в вигляді карти для нанесення географічних об'єктів.

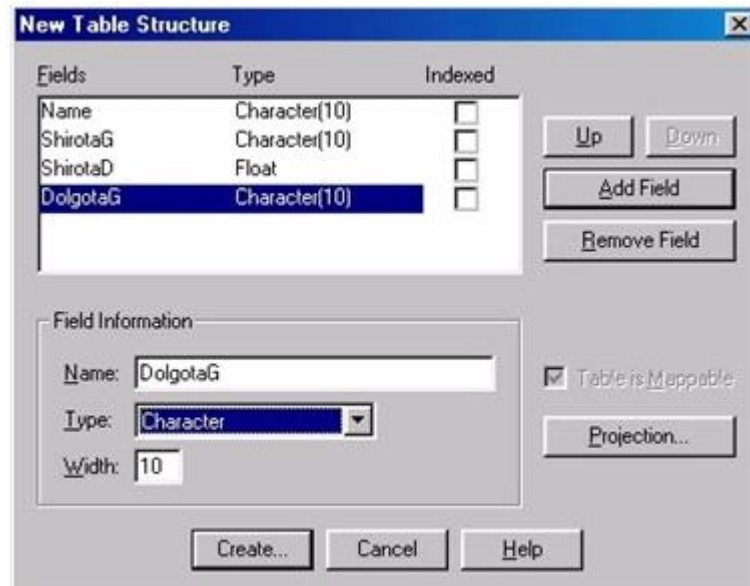


Рис.2. Діалогове вікно другого шагу створення нової таблиці



Рис.3. Діалогове вікно третього шагу створення нової таблиці

Особливу увагу необхідно звернути на кнопку “Projection...” другого вікна. Вона дозволяє вибрати картографічну проекцію шару. Ця проекція має збігатись з проекцією з якої були зняті географічні координати об'єктів (див.п.2). Після закінчення вводу всіх необхідних параметрів та натискання кнопки “Create...” потрібно зберегти новий шар. По закінченні цих процедур система автоматично надасть можливість або заповнити

атрибутивні поля, або нанести географічні об'єкти на карту в залежності від вибору розробника (рис. 1).

Далі розглянемо визначені вище шляхи занесення точкових об'єктів.

Ручне нанесення географічних об'єктів на шар карти за географічними координатами засобами геоінформаційної системи Маючи географічні координати об'єктів, користуючись інструментами системи наносять об'єкти на карту. При цьому можна використовувати такі інструменти (рис. 4): кнопки “Символ” та “Стиль символу” для нанесення точкового об'єкту, перший розділ строки статусу програми (Cursor Location) для визначення місцезнаходження курсору миші (інструменти позначені на малюнку). Для показу координат в географічному форматі можна користуватись діалогом “Опції карти” - група “Представлення координат” (рис.5).

Заповнення атрибутивної частини бази можна проводити як під час нанесення географічних об'єктів, так і після цього. В будь-якому разі для правильного виконання цієї операції необхідно відкрити вікно карти та вікно таблиці (меню Вікно-Нове вікно таблиці) для шару що розробляється, та розташувати їх поруч (рис.6). Позначивши необхідний об'єкт чи то на карті, чи то в таблиці за допомогою миші (відповідні ознаки підкреслені на рисунку 6) можна редагувати його атрибутивні та просторові дані.

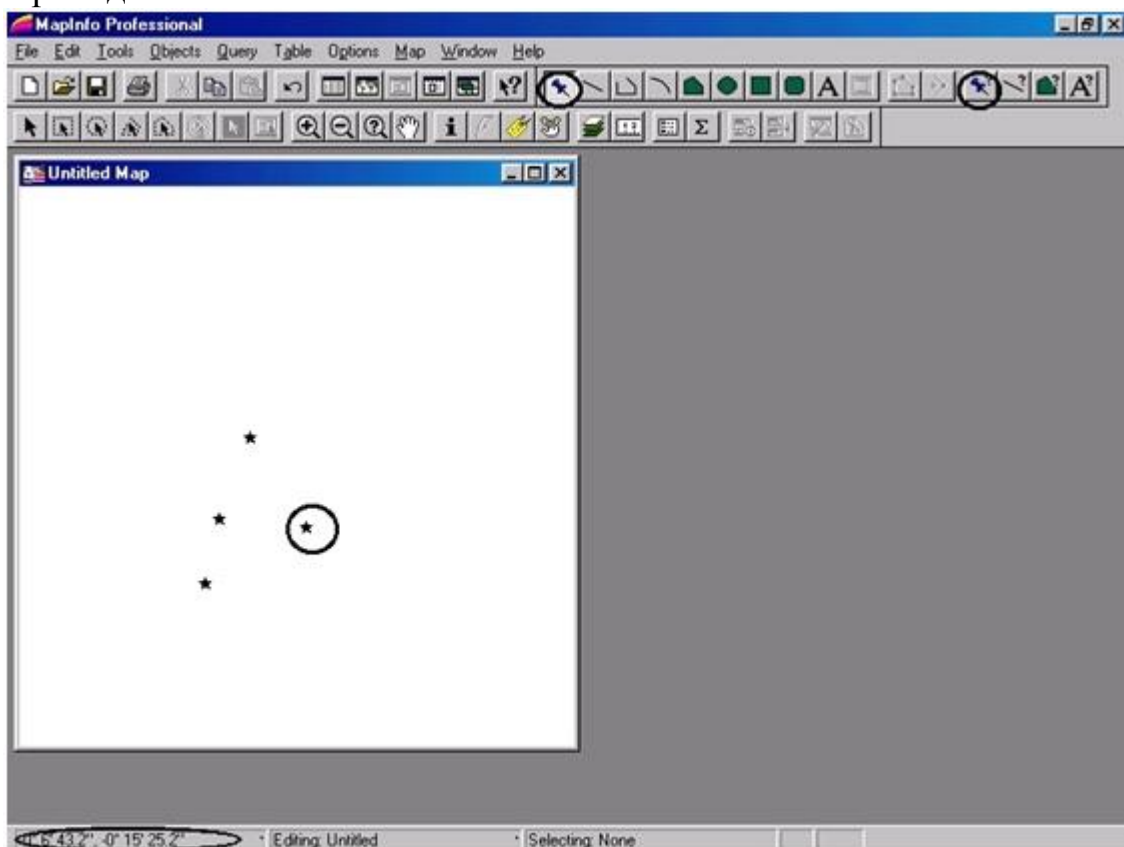


Рис.4. Занесення географічних даних на карту

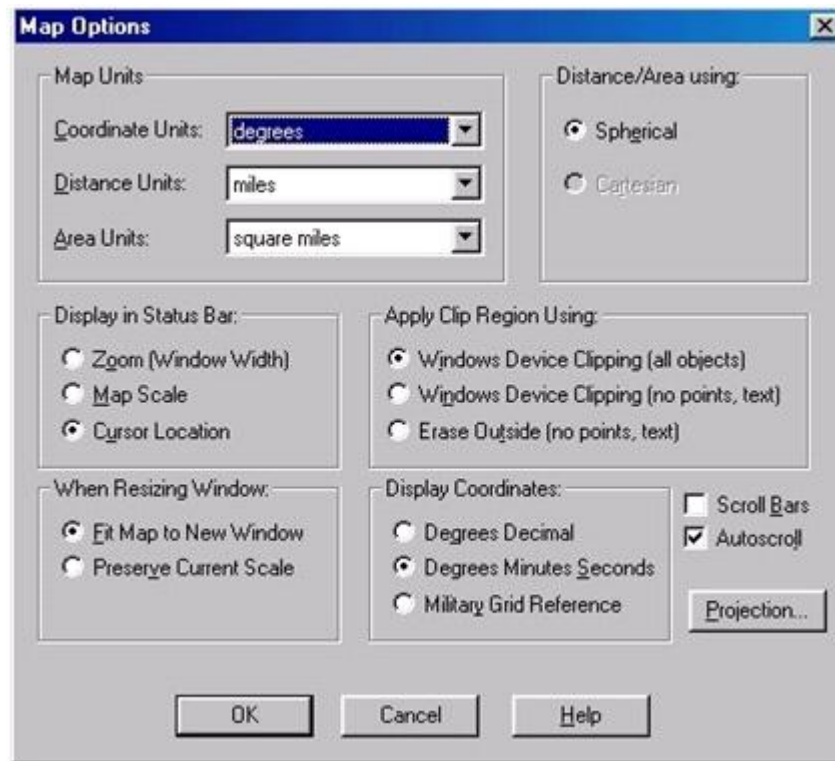


Рис.5. Діалогове вікно “Опції карти”

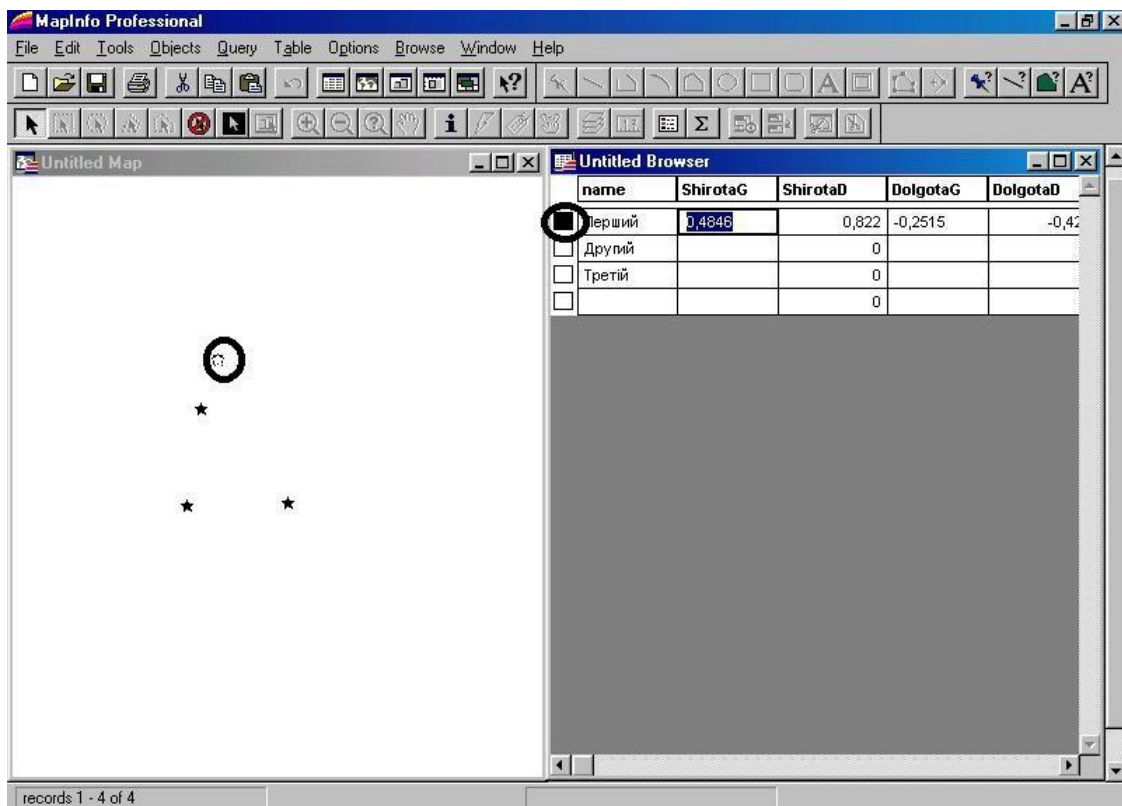


Рис.6. Розташування вікон при редагуванні об’єктів

Для визначення десяткових координат об'єктів можна користуватись формолою (2.1), можливостями системи (діалог “Опції карти”), допоміжними інструментами системи (меню Tools – Tool Manager – Degree Converter), а також системним калькулятором (рис.7).

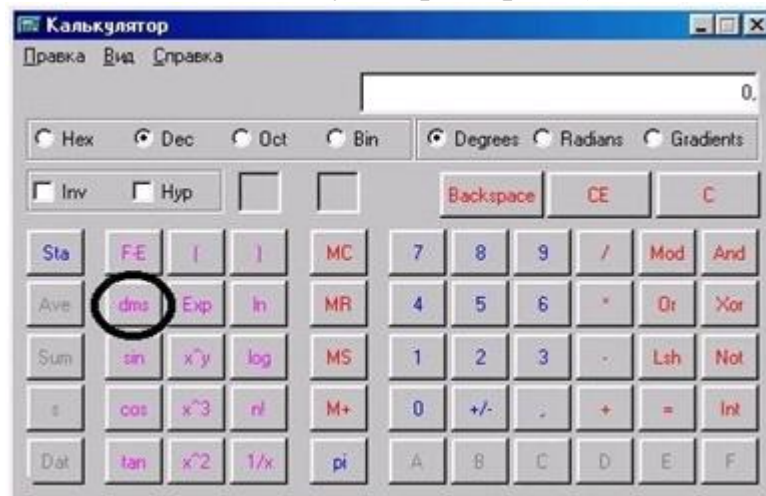


Рис.7. Системний калькулятор

Створення точкових об'єктів засобами системи за наперед занесеними до атрибутивної бази координатами.

Створивши новий шар (п.2.2), та заповнивши необхідні атрибутивні дані (десяткова широта та довгота) розробник може автоматично створити географічні об'єкти користуючись засобами системи (меню Таблиця – Створити точки). Діалогове вікно для створення точок, представлене на рисунку 8, дозволяє задати необхідні параметри для створення картографічних точкових об'єктів за наперед занесеними атрибутивними даними в тій чи іншій таблиці. Насамперед визначається таблиця чи шар для якого будуть створені точкові об'єкти, їх вигляд, картографічна проекція, ім'я стовпчика таблиці в якому занесені координати довготи та широти створюваних об'єктів, та можливі коефіцієнти в разі необхідності перерахунку цих координат.

Слід зауважити, що багато систем представляють географічні координати в звичному для європейської географічної школи вигляді (широта, довгота), але є й такі, наприклад, MapInfo, що використовують представлення американської географічної школи (довгота, широта). Тому при визначенні імен відповідних стовпчиків атрибутів це має бути враховано.

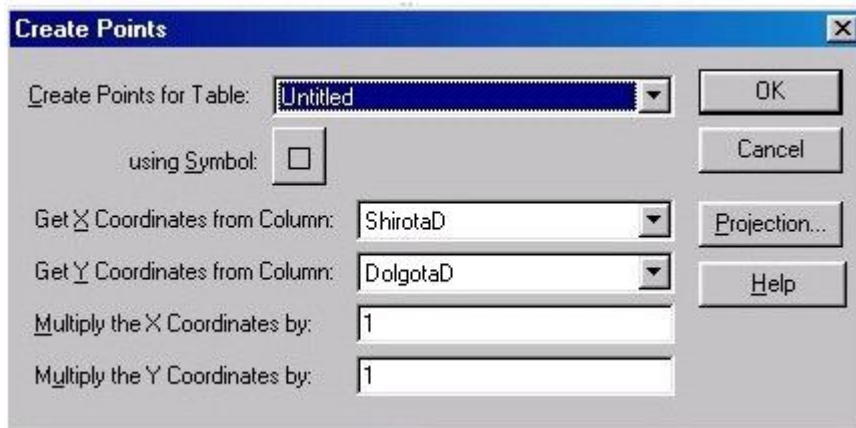


Рис.8. Діалог параметрів для створення точкових об'єктів

2.2 Програмне створення точкових об'єктів. Майже всі потужні геоінформаційні пакети оснащені засобами, що дозволяють значно розширити їх можливості за допомогою створення додаткових інструментів та програм. Ці засоби досить різноманітні: від можливості створення пакетних файлів сценаріїв обробки тих чи інших матеріалів, до повноцінних алгоритмічних мов. Однією з них є MapBasic, призначений для застосування разом з пакетом MapInfo. Це досить потужна мова що дозволяє значно збільшити можливості основного пакету, та створювати на його основі потужні ГІС. В цій частині роботи необхідно за допомогою MapBasic розробити додаток який автоматично створює шар точкових об'єктів за початковими даними, що використовувались в частини 2.1. Початкові дані мають бути занесені в файл формату який задано викладачем та містити такі дані:

1. найменування об'єкту;
2. широта об'єкту географічна;
3. довгота об'єкту географічна.

Результатом роботи додатку має бути географічний шар даних, аналогічний шару, що створено в частині п.2.1.

Опис деяких процедур, функцій, операцій та інструкцій які можуть знадобитися при створенні додатку надано нижче (в квадратних дужках необов'язкові параметри).

CentroidX(), CentroidY() function

Призначення: Повертають відповідно довготу та широту центроїда об'єкта. Синтаксис: CentroidX(*obj_expr*), CentroidY(*obj_expr*)

obj_expr - object – вираз. Значення, що повертається: Float.

Опис: CentroidY() (CentroidX()) функція повертає Y(X)-координату (широта(довгота)) центроїда зазначеного об'єкта.

Open Table statement

Призначення: Відкриває таблицю MapInfo для вводу - виводу.

Синтаксис: Open Table *filespec* [As *tablename*] [Hide] [ReadOnly] [Interactive] [Password *pwd*] [NoIndex]

filespec - визначає таблицю MapInfo, що відкривається;

tablename - "alias(псевдонім)" – ім'я за яким таблиця буде ідентифікуватися; *pwd* - пароль рівня бази даних коли захист бази даних включений. Застосовується тільки до таблиць Access.

Опис: Інструкція Open Table відкриває існуючу таблицю. Відповідає меню File > Open. Таблиця має бути відкрита перш, ніж MapInfo зможе обробляти цю таблицю.

Ім'я файлу, що буде відкритий (зазначений *filespec* параметром) повинне відповідати таблиці, що вже існує. Інструкція Open Table звертається тільки до таблиць MapInfo; щоб використати файли, які перебувають в інших форматах, дивиться інструкції Register Table і Open File.

Якщо інструкція включає пропозицію As, MapInfo відкриває таблицю під ім'ям "псевдоніма", позначеним *tablename* параметром, а не під фактичним ім'ям таблиці. Це впливає на ім'я таблиці яке з'являється, коли користувач вибирає File > Close. Крім того, коли інструкція Open Table визначає ім'я таблиці псевдоніма, наступні MapBasic операції над таблицею (наприклад інструкція Close Table) повинна звернутися до псевдоніма таблиці, а не до постійного ім'я таблиці. Псевдонім таблиці залишається в силі, поки таблиця не закрита. Псевдонім таблиці не зачіпається при перейменуванні таблиці.

Якщо інструкція включає пропозицію Hide, таблиця не буде показуватися ні в яких діалогах, що відображають списки відкритих таблиць (наприклад File > Close діалог). Використовуйте пропозицію Hide, коли треба відкрити таблицю, що повинна залишитися прихованою від користувача.

Якщо інструкція включає пропозицію ReadOnly, користувачеві не дозволяють редагувати таблицю.

Додаткове ключове слово Interactive повідомляє MapBasic запитувати користувача про розташування таблиці, якщо вона не знайдена за зазначеним шляхом. Ключове слово Interactive корисно в ситуаціях, коли невідоме розташування файлів користувача.

Якщо інструкція включає ключове слово NoIndex, індекс MapInfo не буде відновлений для таблиці Access при відкритті.

Close Table statement

Призначення: Закриває відкриту таблицю.

Синтаксис: Close Table *table* [Interactive] *table* - ім'я відкритої таблиці

Опис: Інструкція Close Table закриває відкриту таблицю. Щоб закрити всі таблиці, використовуйте Close All. Якщо таблиця відображена в одному або більше вікнах Grapher або Browser, ці вікна зникають автоматично, коли таблиця закривається. Якщо інструкція Close Table закриває єдину таблицю у вікні Map, вікно закривається. Якщо виключене додаткове ключове слово Interactive, MapBasic закриває таблицю незалежно від того, чи зберегла таблиця редагування; будь-які незбережені редагування відкидаються. Якщо ключове слово Interactive додано, і якщо таблиця не зберегла редагування, MapBasic відображає діалог, що дозволяє користувачеві зберігати або відмовитися від редагувань або скасувати операцію закриття.

Щоб гарантувати незбереження останнього редагування відкиньте ключове слово Interactive або запустіть інструкцію RollBack перед викликом Close Table. Щоб гарантувати збереження редагування викличте інструкцію Commit перед інструкцією Close Table. Щоб визначити, чи не зберегла таблиця редагування, викликають TableInfo (table, TAB_INFO_EDITED) функцію.

Create Table statement

Призначення: Створює нову таблицю.

Синтаксис: Create Table *table*

(*column columntype* [, ...]) [File *filespec*]

[{ Type NATIVE |

Type DBF [CharSet *char_set*] |

Type Access Database *database_filespec* Table *tablename*

[Password *pwd*] [CharSet *char_set*]

}]

[Version *version*]

table - ім'я таблиці;

column - ім'я створюваного стовпця. Ім'я стовпця може бути до 31 символу довжиною, і може містити символи, числа, і символ підкреслення (_). Ім'я стовпця не може починатися із чисел.;

filespec - визначає де створювати .TAB, .MAP, і .ID файли (і у випадку Access, AID файли). Якщо пропозицію File опущено, файли будуть створені в поточному каталозі;

char_set - ім'я набору символів, типу "WindowsLatin1";

database_filespec - рядок, що ідентифікує існуючу базу даних Access. Якщо зазначена база даних не існує, MapInfo створює новий .MDB файл Access; *tablename* - рядок, що вказує ім'я таблиці, з яким вона з'явиться в Access;

pwd - пароль рівня бази даних, коли захист бази даних включений;
version - 100 (для створення таблиці, що може читатися більше ранніми версіями MapInfo) або 300 (MapInfo 3.0 формат). Не застосовується при створенні таблиці Access; версія таблиці Access обробляється DAO.

columnntype - тип даних, пов'язаний зі стовпцем. Кожний *columnntype* визначається в такий спосіб:

Char(width) | Float | Integer | SmallInt | Decimal(width, decplaces) | Date | Logical width - указує необхідну довжину поля (не ставиться до всіх типів полів). Поля Char можуть мати ширину до 254 символів.

decplaces - указує число десяткових місць, у поле Decimal

Опис: Інструкція Create Table створює нову порожню таблицю. Додатковий *filespec* визначає, де створити нову таблицю. Якщо ніяка *filespec* пропозиція не використовується, таблиця створюється в поточному каталозі або папці. Додаткова пропозиція Type визначає формат даної таблиці. Значення за замовчуванням NATIVE, але може бути DBF. Формат NATIVE займає меншу кількість дискового простору ніж формат DBF, але формат DBF може читатися в будь-якому dBASE-сумісному адміністраторі бази даних. Пропозиція CharSet визначає набір символів. *Char_set* параметр повинен бути строковою константою, типу "MacRoman" або "WindowsLatin1". Якщо ніяка пропозиція CharSet не визначена, MapBasic використає заданий за замовчуванням набір символів для апаратної платформи. Тип стовпця SmallInt резервує два байти для кожного значення; таким чином, стовпець може містити значення від -32767 до +32767. Тип стовпця Integer резервує чотири байти; таким чином, стовпець може містити значення від -2147483647 до +2147483647. Пропозиція Version управляє форматом таблиці; ця пропозиція не застосовується, коли створюється таблиця Access. Якщо зазначено Version 100, MapInfo створює таблицю у форматі, що може читатися версіями MapInfo раніше чим 3.0. Якщо Version 300, MapInfo створює таблицю у форматі, використовуваному MapInfo 3.0. Зверніть увагу, що область і об'єкти ламаної лінії, що мають більше чим 8,000 вузлів і об'єкти багатосегментної ламаної лінії вимагають версії 300. Якщо опущено пропозицію Version, таблиця буде створена у форматі версії 300.

Register Table statement

Призначення: Будує таблицю MapInfo з електронної таблиці, бази даних, або текстового файлу (але не з растрового зображення).

Синтаксис: Register Table *source_file*

{ Type "NATIVE" |

Type "DBF" [Charset *char_set*] |

```
Type "ASCII" [ Delimiter delim_char ] [ Titles ] [ CharSet char_set ] |
Type "WKS" [ Titles ] [ Range range_name ] |
Type "XLS" [ Titles ] [ Range range_name ] |
Type "Access" Table table_name [ Password pwd ] [ CharSet char_set ] }
[ Into destination_file ]
```

source_file - рядок, що визначає ім'я існуючої бази даних, електронної таблиці, або текстового файлу. Якщо реєструється таблиця Access, цей параметр повинен ідентифікувати існуючу базу даних Access;

char_set - ім'я набору символів;

delim_char - визначає символ, що використовується як роздільник стовпців. Якщо використовується як роздільник Tab (табуляція), визначте 9. Якщо файл використовує коми, визначте 44;

range_name - рядок, що вказує названий діапазон (наприклад "MyTable") або діапазон осередків (наприклад Excel діапазон може бути визначений як " Sheet1! R1C1:R9C6 " або як " Sheet1! A1:F9 ");

table_name - рядок, що ідентифікує ім'я таблиці Access;

pwd - пароль рівня бази даних коли захист бази даних включена;

destination_file - визначає ім'я, що надається таблиці MapInfo (.TAB файл). Цей рядок може включати шлях; якщо вона не включає шлях, файл створюється в тому ж каталозі, що й вихідний файл.

Опис: Перш, ніж використовувати "нерідний" файл (наприклад dBASE файл) в MapInfo, його треба зареєструвати. Як тільки операція Register Table побудувала файл таблиці, можна звертатися до нього як до таблиці MapInfo. Інструкція Register Table не копіює та не змінює первісний файл даних. Замість цього, вона переглядає дані, визначає типи даних стовпців, і створює окремий файл таблиці. Зверніть увагу: таблиця не відкривається автоматично. Щоб відкрити таблицю, використовуйте інструкцію Open Table. Кожний файл даних повинен бути зареєстрований тільки один раз. Як тільки операція Register Table побудувала відповідний файл таблиці, наступні сеанси MapInfo просто відкривають таблицю, а не повторюють операцію Register Table.

Пропозиція Type визначає, тип даних початкового файлу. Воно складається із ключового слова Type, супроводжуваного однією з наступних символічних констант: NATIVE, DBF, ASCII, WKS, XLS, або Access. Інша інформація необхідна для підготовки деяких типів таблиць.

Пропозиція CharSet визначає набір символів. *Char_set* параметр повинен бути рядком типу "MacRoman" або "WindowsLatin1". Якщо Ви опускаєте пропозицію CharSet, MapInfo використає заданий за замовчуванням набір символів для апаратної платформи, що використовується під час виконання.

Пропозиція Delimiter супроводжується рядком, що містить символ

роздільника. Заданий за замовчуванням роздільник - TAB (табуляція). Пропозиція Titles вказує, що рядок перед діапазоном даних в worksheet повинен бути використаний як рядок к заголовками стовпців. Пропозиція Range дозволяє використати визначення йменованих діапазонів. Пропозиція Into використовується, щоб змінити ім'я таблиці або розташування .TAB файлу. За замовчуванням, вона буде названа тим же ім'ям, що й файл даних, і збережена в тому ж каталозі. Однак, при читанні пристрою типу CD-ROM, Ви повинні зберегти .TAB файл в іншому місці.

Інструкція Register Table не може використовуватись за для того, щоб зробити растровий файл зображення (наприклад. Photo.GIF) MapInfo таблицею (Photo.TAB). Щоб побудувати таблицю, засновану на растровому зображенні, програма повинна побудувати файл таблиці використовуючи інструкцій файлового вводу / виводу типу Open File і Print #.

Коли реєструється таблиця Access, MapInfo перевіряє автоінкриментний стовпець із унікальним індексом. Якщо вже є такий стовпець, MapInfo реєструє стовпець в .TAB файлі. Стовпець тільки для читання. Якщо таблиця Access не має такого стовпця, MapInfo змінює таблицю Access, додаючи стовпець за ім'ям MapInfo_ID з автоінкриментним типом. У цьому випадку, такий стовпець не відображається в MapInfo.

Типи даних Access транслюються в найближчий тип даних MapInfo Спеціальні типи даних Access, типу OLE об'єктів і інших бінарних полів недоступні для редагування в MapInfo.

Alter Table statement

Призначення: Змінює структуру таблиці. Не може використовуватись для зв'язаних таблиць.

Синтаксис: Alter Table *table* (
[Add *columnname columntype* [, ...]] [Modify *columnname columntype* [,
..]]
[Drop *columnname* [, ...]]
[Rename *oldcolumnname newcolumnname* [, ...]] [Order *columnname*,
columnname [,...]]
)
[Interactive]

table – ім'я відкритої таблиці;

columnname - ім'я стовпця; ім'я стовпця може бути до 31 символа довжиною, і може містити символи, числа, і символ підкреслення; ім'я стовпця не може по- чинатися із числа;

columntype - тип даних стовпця таблиці (включаючи ширину поля

якщо буде потреба);

oldcolumnname - попереднє ім'я стовпця, що буде перейменований;

newcolumnname - нове ім'я стовпця, що буде перейменований;

Опис: Інструкція Alter Table дозволяє змінювати структуру відкритої таблиці, дозволяючи додавати стовпці, змінювати ширину або тип даних, видаляти, перейменовувати стовпці. Зверніть увагу: перед використанням інструкції Alter Table редагування таблиці мають бути або збережені, або відкинуті. Кожен тип стовпця має бути одним з наступних: Integer, SmallInt, Float, Decimal (size, decplaces), Char (size), Date, or Logical.

Включенням пропозиції Add в інструкції Alter Table, можна додавати нові стовпці до таблиці. Включенням пропозиції Modify, можна змінювати тип даних існуючих стовпців. Пропозиція Drop дозволяє видаляти стовпці, у той час як пропозиція Rename дозволяє змінювати імена існуючих стовпців. Пропозиція Order дозволяє визначати порядок стовпців. У цілому, інструкція Alter Table може мати до п'яти пропозицій. Зверніть увагу, що кожне із цих п'яти пропозицій може працювати зі списком стовпців; таким чином, з однією інструкцією Alter Table, можна зробити всі структурні зміни.

Пропозиція Order впливає на порядок стовпців, а не порядок рядків у таблиці. Порядок стовпця визначає відносні позиції стовпців, коли переглядається таблиця; перший стовпець з'являється в лівому краї вікна Browser, і останній стовпець з'являється в правому краї. Точно так само перший стовпець таблиці з'являється нагорі вікна інструмента Info.

Якщо прикладна програма MapBasic запускає інструкцію Alter Table, що впливає на таблицю, що має темо поля, поля темо будуть загублені без будь-якого попередження. Якщо структура таблиці змінена через інтерфейс користувача MapInfo (вибором структури таблиці з Table > Maintenance меню), MapInfo попереджає користувача про втрату темо поля. Однак, коли структура таблиці змінюється програмою MapBasic, ніякого попередження не з'являється.

Інструкція Alter Table може видаляти шари карти з вікна Map, можливо завдаючи втрату тим або косметичним об'єктам. Якщо включене ключове слово Interactive, MapInfo запитує користувача зберігати теми й/або косметичні об'єкти (якщо теми або косметичні об'єкти можуть бути загублені).

Add Column statement

Призначення: Додає новий, тимчасовий стовпець до відкритої таблиці, або модифікує існуючий стовпець даними з іншої таблиці.

Синтаксис:

Add Column *table* (*column* [*datatype*])

```
{ Values const [ , const ... ] | From source_table  
Set To expression  
[ Where { dest_column = source_column | Within | Contains | Intersects } ]  
[ Dynamic ] }
```

table - ім'я таблиці, до якої буде доданий стовпець;

column - ім'я нового стовпця, що додається;

datatype - тип даних стовпця, визначається як Char(width), Float, Integer, SmallInt, Decimal(width, decimal_places), Date або Logical; Якщо не зазначений, значення за замовчуванням Float;

source_table - ім'я іншої відкритої таблиці;

expression - вираз, для обчислення значення, що зберігається в новому стовпці; цей вираз зазвичай використовує дані з *source_table*, і може включати агрегатні функції;

dest_column - ім'я стовпця з таблиці адресата (*table*);

source_column - ім'я стовпця з *source_table*.

Dynamic визначає динамічний (гарячий) обчислюваний стовпець: якщо включене це ключове слово, то наступні зміни, зроблені у вихідній таблиці автоматично застосовуються до таблиці адресата.

Опис: Інструкція Add Column створює тимчасовий новий стовпець для існуючої таблиці MapInfo. Новий стовпець постійно не буде зберігатися на диск. Однак, якщо тимчасовий стовпець заснований на основних таблицях, і якщо Ви зберігаєте робочий простір, у той час як тимчасовий стовпець перебуває у використанні, робочий простір буде включати інформацію про тимчасовий стовпець, так, щоб він був відновлений, якщо робочий простір відновиться. Щоб додавати постійний стовпець до таблиці, використайте інструкції Alter Table і Update.

Використовуючи пропозицію Values, Ви можете визначити відділений комою список явних значень, щоб зберегти їх в новому стовпці.

Якщо визначено пропозицію From замість пропозиції Values, MapBasic одержує значення для нового стовпця з окремої таблиці (*source_table*). Обидві таблиці повинні бути відкриті. MapInfo з'єднує ці дві таблиці. Щоб визначити, як ці дві таблиці з'єднані, включити додаткову пропозицію Where. Якщо Ви опускаєте пропозицію Where, MapInfo автоматично пробує з'єднувати таблиці, використовуючи найбільш підходящий метод. Якщо обидві таблиці містять об'єкти карти, пропозиція Where може визначити географічне об'єднання. Наприклад, якщо визначена пропозиція Where Contains, MapInfo створює об'єднання, перевіряючи, чи містять об'єкти з *source_table* об'єкти з таблиці, що змінюється.

Наступний приклад додає стовпець "County" у таблицю "Stores". Новий стовпець буде містити імена країн, які витягнуті з окремої таблиці

областей країн:

Add Column

stores(county char(20)) 'add "county" column

From counties 'derive data from counties table...

Set to cname 'using the counties table's "cname" column Where Contains
'join: where a county contains a store site

Where Contains метод підходить, коли додається стовпець до таблиці точкових об'єктів, і друга таблиця представляє об'єкти, які містять точки.

Create Map statement

Призначення: Змінює структуру таблиці, роблячи таблицю картографічною.

Синтаксис: Create Map

For *table*

[CoordSys...]

table - ім'я відкритої таблиці;

Опис: Інструкція Create Map робить відкрити таблицю картографічною, так, щоб вона могла бути відображена у вікні Map. Ця інструкція не відкриває нове вікно Map. Щоб відкрити нове вікно Map, використовуйте інструкцію Map.

Не потрібно виконувати інструкцію Create Map щодо таблиці, яка вже є картографічною; таке виконання видалить всі об'єкти карти з таблиці. Якщо до таблиці вже прикріпили карту, і потрібно надовго замінити проекцію карти, використовуйте інструкцію Commit Table As. Також, якщо потрібно тимчасово змінити проекцію, у якій карта відображена, використовуйте інструкцію Set Map із пропозицією CoordSys.

Інструкція Create Map не працює на зв'язаних таблицях. Щоб зробити зв'язану таблицю картографічною, використовуйте інструкцію Server Create Map.

Якщо опущено пропозицію CoordSys, таблиця буде використовувати завдану за замовчуванням Земну (Earth) систему координат (координати, збережені як довгота/широта). Зверніть увагу, що пропозиція CoordSys впливає на точність карти. Пропозиція CoordSys включає пропозицію Bounds, що встановлює мінімальні і максимальні координати, які можуть бути збережені в карті. Якщо опущено пропозицію Bounds, MapInfo використає завдані за замовчуванням границі, які охоплюють повну Землю (в цьому випадку, точність координат приводиться до однієї мільйонної градуса, або приблизно 4 дюймів). Якщо відомо заздалегідь, що карта буде обмежена кінцевою областю, можна збільшувати точність координат карти, визначаючи границі, які обмежують карту в тій області.

Set Table statement

Призначення: Конфігурує різні параметри настроювання відкритої таблиці.

Синтаксис: Set Table *tablename*

```
[ FastEdit { On | Off } ] [ Undo { On | Off } ] [ ReadOnly ]  
[ Seamless { On | Off } [ Preserve ] ] [ UserMap { On | Off } ]  
[ UserBrowse { On | Off } ] [ UserClose { On | Off } ] [ UserEdit  
  { On | Off } ]  
[ UserRemoveMap { On | Off } ] [ UserDisplayMap { On | Off } ]
```

Опис: Інструкція Set Table управляє параметрами настроювання, які визначають, як і чи може таблиця бути відредагована. Set Table може позначити таблицю тільки для читання (так, щоб користувачеві не дозволити робити зміни в таблиці), активізувати або дезактивізувати спеціальні режими редагування, які відключають безпечні механізми заради поліпшення ефективності редагування.

Commit Table statement

Призначення: Зберігає недавні редагування на диск, або зберігає копію таблиці.

Синтаксис:

Commit Table *table*

```
[ As filespec  
[ Type {NATIVE |  
DBF [ Charset char_set ] |  
Access Database database_filespec Table tablename  
[ Password pwd ] [ CharSet char_set ]  
}  
]  
[ CoordSys... ]  
[ Version version ] ]  
[ { Interactive | Automatic commit_keyword } ]
```

table - ім'я таблиці, що зберігається;

filespec - специфікація файлу (довільно, включаючи шлях каталогу);

char_set - ім'я набору символів;

database_filespec - рядок, що вказує шлях та ім'я бази даних, де таблиця MapInfo буде збережена. Якщо база даних не існує, MapInfo створює нову базу даних;

tablename - ім'я таблиці в базі Access;

pwd - пароль рівня бази даних, коли захист бази даних включений;
CoordSys - пропозиція системи координат;

version - 100 (щоб створити таблицю, що може читатися більш ранніми версіями MapInfo) або 300 (MapInfo 3.0 формату) для не Access таблиць. Для таблиць Access, версія - 410.

commit_keyword - одне з наступних ключових слів: NoCollision, ApplyUpdates, DiscardUpdates.

Опис: Якщо пропозиція As не визначена, інструкція Commit зберігає будь-які обробки, що чекають, редагування в таблиці. Це аналогічно вибору меню File > Save. Інструкція Commit, що включає пропозицію As, робить теж саме що й клієнтський користувач, коли вибирає File > Save Copy As. Пропозиція As може використовуватися, щоб зберегти таблицю з іншим ім'ям, каталогом, типом файлу, або проекцією.

Щоб зберегти таблицю під новим ім'ям, задайте нове ім'я в рядку *filespec*. Щоб зберегти таблицю в іншому каталозі, визначите шлях каталогу в рядку *filespec*. Щоб зберегти таблицю, використовуючи новий типфайла, включіть пропозицію Type у межах пропозиції As. За замовчуванням, тип нової таблиці NATIVE, але може також бути збережений як DBF.

Пропозиція CharSet визначає набір символів. *Char_set* параметр повинен бути строковою константою, типу "MacRoman" або "WindowsLatin1". Якщо пропозиція CharSet не визначена, MapBasic використає завданий за замовчуванням набір символів для апаратної платформи, що перебуває у використанні під час виконання.

Щоб зберегти таблицю, використовуючи іншу систему координат або проекцію, включіть пропозицію CoordSys у межах пропозиції As. Зверніть увагу, що тільки картографічна таблиця може мати систему координат або проекцію.

Пропозиція Version управляє форматом таблиці. Якщо Ви визначаєте 100, MapInfo зберігає таблицю у форматі, що читає більше ранніми версіями MapInfo. Якщо Ви визначаєте Версію 300, MapInfo зберігає таблицю у форматі, використовуюваному MapInfo 3.0. Зверніть увагу, що область і об'єкти ламаної лінії, що мають більше чим 8,000 вузлів і об'єкти багатосегментної ламаної лінії вимагають версії 300. За замовчуванням таблиця зберігається у форматі версії 300.

Попередження: Якщо прикладна програма MapBasic запускає Commit Table...As інструкцію, що впливає на таблицю, що має поля memo, поля memo не будуть збережені в новій таблиці. Ніяке попередження не буде відображено. Якщо таблиця збережена в новій таблиці через інтерфейс користувача MapInfo (вибираючи File > Save Copy As), MapInfo попереджає користувача про втрату полів memo. Однак, коли таблиця збережена з новим ім'ям MapBasic програмою, ніяке попередження не

з'являється.

Open File statement

Призначення: Відкриває файл для вводу/виводу.

Синтаксис: Open File *filespec*

[For { Input | Output | Append | Random | Binary }] [Access { Read
| Write | Read Write }]

As [#] *filenum*

[Len = *recordlength*]

[ByteOrder { LOWHIGH | HIGHLOW }]

[CharSet *char_set*]

[Filetype *macfiletype*]

filespec - String: ім'я файлу, що відкривається;

filenum - Integer номер, асоційований з відкритим файлом; цей номер використовується в наступних операціях (наприклад Get, Put)

recordlength - визначає число символів у запису, включаючи будь-які використовувані маркери кінця рядка; застосовується тільки при довільному доступі; *char_set* - ім'я набору символів;

macfiletype - рядок із чотирьох символів, що вказує тип файлу для Макінтош

(наприклад "Mlwo");

Обмеження: Не можна запускати інструкцію Open File у вікні MapBasic.

Опис: Інструкція Open File відкриває файл для читання з і/або запису інформації у файл.

В MapBasic, є важливе розходження між файлами й таблицями. MapBasic забезпечує один набір інструкцій для використання таблиць (наприклад Open Table, Fetch, і Select) і інший набір інструкцій для використання інших файлів (наприклад Open File, Get, Put, Input #, Print #).

Пропозиція For визначає який доступ до файлу вводу/виводу виконати: послідовний, довільний, або бінарний. За замовчуванням, файл відкривається у довільному режимі.

Close File statement

Призначення: Закриває відкритий файл.

Синтаксис: Close File [#] *filenum*

filenum - Integer номер, що ідентифікує файл

Опис: Інструкція Close File закриває файл, що був відкритий інструкцією Open File.

Зверніть увагу, що Open File і Close File працюють взагалі з файлами,

а не з MapInfo таблицями. MapBasic забезпечує окремий набір інструкцій (наприклад Open Table) для керування MapInfo таблицями.

Get statement

Призначення: Читає з файлу відкритого у бінарному або довільному (випадковому) режимі доступу.

Синтаксис: Get [#] *filenum* , [*position*] , *var_name filenum* - номер файлу, відкритого інструкцією Open file; *position* позиція у файлі з якої починається читання;

var_name – ім'я змінної, у якій MapBasic збереже результати;

Опис: Інструкція Get читає з відкритого файлу. На поводження інструкції Get і набору параметрів, які вона очікує, впливають параметри, зазначені в попередній інструкції Open File.

Якщо інструкція Open File визначила довільний доступ до файлу, пропозиція *Position* інструкції Get може використовуватись, щоб вказати дані для читання. Коли файл відкритий, позиція файлу вказує на перший запис файлу (запис 1). Get автоматично збільшує позицію файлу, і в такий спосіб пропозиція *Position* не повинне використовуватися, якщо виконується послідовний доступ. Однак, можна використовувати пропозицію *Position*, щоб установити позицію запису перш, ніж запис буде прочитано.

Якщо інструкція Open File визначила бінарний доступ до файлу, одна змінна може читатися один раз. Які дані читаються, залежить від формату порядку байта файлу й *var_name* змінної, використовуваної, щоб зберегти результати. Якщо тип змінної цілочисельний, то 4 байти бінарного файлу будуть прочитані.

Input # statement

Призначення: Читає дані з файлу, і зберігає дані в змінній.

Синтаксис: Input # *filenum*, *var_name* [, *var_name* ...] *filenum* - номер файлу, відкритого Open File;

var_name - ім'я змінної;

Опис: Input # читає дані з файлу, що був відкритий у послідовному режимі (наприклад. INPUT режим), і зберігає дані в одну або більше MapBasic змінну. Input # читає дані (до наступного кінця рядка) у змінну(i), позначену *var_name* параметром (ами). MapInfo обробляє коми й символи кінця рядка як роздільники полів. Щоб читати повну лінію тексту в окрему рядкову змінну, використовуйте Line Input #.

MapBasic автоматично перетворює дані до типу змінної(их). При читанні даних у рядкову змінну, Input # обробляє порожню лінію як порожній рядок. При читанні даних у числову змінну, Input # обробляє

порожню лінію як нульове значення.

Put statement

Призначення: Пише зміст MapBasic змінної у відкритий файл.

Синтаксис: Put [#] *filenum*, [*position*], *var_name filenum* - номер файлу, відкритий Open File;

position - позиція файлу, з якої записується змінна у файл (не встановлюється при записі у файл послідовного доступу);

var_name - ім'я змінної, котра містить дані, які будуть записані.

Опис: Інструкція Put записує у відкритий файл.

Зверніть увагу: якщо інструкція Open File визначила режим послідовного доступу (OUTPUT або APPEND), використовуйте Print # или Write # замість Put.

Якщо інструкція Open File визначила файл довільного доступу, пропозиція *Position* інструкції Put може використовуватись, щоб вказати, який запис у файлі переписується. Коли файл відкривається, позиція файлу вказує на перший запис файлу (запис 1).

Якщо інструкція Open File визначила доступ до бінарного файлу, одна змінна може бути записана один раз. Послідовність байтів, записаних у файл залежить від апаратної платформи. Число записаних байтів залежить від типу змінної, дивіться Get.

Параметр *Position* встановлює покажчик файлу на певну позицію у файлі. Коли файл відкритий, позиція ініційована в 1 (початок файлу). Коли Put виконаний, позиція збільшується на число записаних байтів. Якщо пропозиція *Position* не використовується, Put просто записує в поточну позицію файлу.

Якщо файл був відкритий у бінарному режимі, інструкція Put не може ви- значити змінну змінної довжини; будь-яка строкова змінна, використовувана в інструкції Put повинна бути фіксованої довжини. Якщо файл був відкритий у довільному режимі, інструкція Put не може визначити змінну фіксованої довжини, що є більш довгою ніж довжина запису файлу.

Print # statement

Призначення: Пише дані у файл відкритий у послідовному режимі (Output або Append).

Синтаксис: Print # *file_num* [, *expr*]

file_num - номер файлу, відкритого інструкцією Open File;

expr - вираз, записуваний у файл.

Опис: Print # пише дані у відкритий файл. Файл повинен бути

відкритий, у послідовному режимі виводу (Output або Append). *File_num* параметр відповідає номеру, зазначеному в пропозиції As інструкції Open File. MapInfo пише вираз *expr*. Щоб зберігати відділений комами список виразів на кожній лінії файлу, використовуйте Write # замість Print #.

Write # statement

Призначення: Пише дані у відкритий файл

Синтаксис: Write # *file_num* [, *expr ...*] *file_num* - номер відкритого файлу;

expr - вираз, записуваний у файл.

Опис: Write # пише дані у відкритий файл. Файл, повинен бути відкритий у послідовному режимі, що дозволяє модифікацію файлу (Output або Append). *File_num* параметр відповідає номеру, зазначеному в пропозиції As інструкції Open File.

Якщо інструкція включає відділений комами список виразів, MapInfo автоматично вставляє коми у файл, щоб відокремити елементи. Якщо інструкція не включає ніякі вирази, MapInfo пише порожню лінію у файл. Write # автоматично маркірує рядок лапками. Щоб писати текст без маркування, використовуйте інструкцію Print #. Використайте Input #, щоб читати файли, які були створені Write #.

EOF() function

Призначення: Повертає TRUE, якщо MapBasic пробував читати за кінцем файлу, інакше FALSE.

Синтаксис: EOF(*filenum*)

filenum номер файлу, відкритого Open File

Значення, що повертає: Logical

Опис: EOF() функція повертає логічне значення, що вказує, чи існує стан End- Of-File. Ціле число *filenum* представляє номер відкритого файлу. Якщо інструкція Get пробує читати за кінцем зазначеного файлу, EOF() повертає значення TRUE, інакше повертає значення FALSE. Зверніть увагу, що EOF() функція працює з відкритими файлами; коли потрібно перевірити поточну позицію відкритої таблиці, використовуйте функцію EOT().

EOT() function

Призначення: Повертає TRUE, якщо MapBasic досяг кінця зазначеної таблиці, інакше FALSE.

Синтаксис: EOT (*table*)

table - ім'я відкритої таблиці. Значення, що повертає: Logical

Опис: EOT() функція повертає TRUE або FALSE указуючи, чи пробував MapInfo читати за кінцем зазначеної таблиці.

Fetch statement

Призначення: Установлює позицію курсору таблиці (тобто встановлює рядок поточним рядком).

Синтаксис: Fetch { First | Last | Next | Prev | Rec *n* } From *table n* - число прочитаних записів;

table - ім'я відкритої таблиці.

Опис: Використовуйте інструкцію Fetch, щоб встановити запис у відкритій таблиці. Інструкція Fetch, розміщає курсор таблиці в деяку позицію рядка в таблиці; що визначає яка із записів у таблиці є "поточним" записом.

Зверніть увагу: термін "курсор" використовується тут, щоб показати позицію рядка в таблиці. Це не має ніякого відношення до екранного курсору миші.

Після того, як Ви запускаєте інструкцію Fetch, Ви можете одержати дані з поточного рядка, використовуючи одне з наступних виражень:

Fetch First інструкція встановлює курсор у перший не вилучений рядок у таблиці.

Fetch Last інструкція встановлює курсор в останній не вилучений рядок у таблиці.

Fetch Next інструкція пересуває курсор уперед на наступний не вилучений рядок.

Fetch Prev інструкція переміщає курсор назад у попередній не вилучений рядок. Fetch Rec *n* інструкція встановлює курсор у певний рядок, навіть якщо цей рядок вилучений. Зверніть увагу: Якщо зазначений запис вилучений, інструкція генерує помилку **404** під час виконання програми.

Різні операції MapInfo і MapBasic (наприклад Select, Update, та ті що змінюють екран) автоматично скидають поточний рядок. Відповідно, інструкції Fetch повинні бути запущені саме перед будь-якими інструкціями, які мають на увазі знання поточного рядка.

Після виклику інструкції Fetch можна викликати EOT() функцію, щоб визначити, чи вибрані фактичні рядки. Якщо інструкція Fetch помістила курсор у фактичний рядок, EOT() функція поверне FALSE (немає кінця таблиці). Якщо інструкція Fetch спробувала розмістити курсор за останнім рядком, EOT() функція поверне TRUE (є кінець таблиці); тому немає ніякого "поточного рядка").

Контрольні питання

1. Основні типи графічних примітивів для представлення географічних об'єктів.
2. Що таке десяткові градуси.
3. Як визначаються десяткові градуси.
4. Алгоритм перерахунку десяткових градусів в звичайні.
5. Етапи створення шару географічних даних
6. Функції MapBasic для створення точкових об'єктів.

Варіанти завдань

1. Створити в MapInfo шар для занесення населених пунктів (об'єктів) країн, та визначити перелік їх атрибутивних даних. Занести п'ять видів атрибутивної інформації яки мають бути присутні в шарі обов'язково, а саме: найменування об'єкту; широта об'єкту географічна; широта об'єкту десяткова; довгота об'єкту географічна; довгота об'єкту десяткова. Растрове зображення (рис.9) ділянки карти, з вказівкою населених пунктів, представлено у файлах *Europe(z8).tab* и *Europe(z8).jpg*.

2. Розробити за допомогою MapBasic додаток, який автоматично створює шар точкових об'єктів - населених пунктів, за початковими даними, а саме: найменування об'єкту; широта об'єкту географічна; широта об'єкту десяткова; довгота об'єкту географічна; довгота об'єкту десяткова. Початкові дані мають бути занесені в файл формату який задано викладачем та містити такі дані: найменування об'єкту; широта об'єкту географічна; довгота об'єкту географічна.

Растрове зображення (рис.9) ділянки карти, з вказівкою населених пунктів, представлено у файлах *Europe(z8).tab* и *Europe(z8).jpg*.



Рис. 9 Карта территорий

Таблица 1. Варианты заданий

№ варианта	Наименование территории (государства)	Примечание
1	Румыния	Бухарест
2	Угорщина	Будапешт
3	Чехия	Прага
4	Италия	Рим
5	Франция	Париж
6	Німеччина	Берлін
7	Испания	Мадрид
8	Греция	Афины
9	Польша	Варшава
10	Австрия	Відень
11	Болгария	София
12	Туреччина	Стамбул

Завдання №2

Тема: Характеристики геоінформаційних об'єктів

Мета роботи

В геоінформаційних базах даних крім атрибутивних даних, які зберігаються як правило у достатньо відомих форматах СУБД, зберігаються також і географічні дані. Різні ГІС системи зберігають ці дані застосовуючи різні методи та підходи. Одні використовують звичайні формати СУБД, дозволяючи користувачу безпосередньо звертатись до цих даних та обробляти їх, інші використовують свої формати збереження географічних даних, надаючи користувачеві можливості їх обробки лише засобами системи. Але всі геосистеми оснащені засобами обробки не тільки атрибутивних, а й географічних даних. Різноманітні характеристики географічних об'єктів, зокрема, просторові є визначальними при проведенні географічного, геостатистичного та інших видів аналізу в використанні просторових характеристик. До того ж, кожен тип географічного об'єкту (точка, лінія (ламана), полігон, центроїд, мітка) має свій набір характеристик, в тому числі й просторових.

Метою даної роботи є ознайомлення студентів з можливостями та особливостями роботи з різноманітними географічними об'єктами та визначенням їх геоінформаційних характеристик.

Завдання до роботи

Завдання – надати необхідні знання та виробити навички у студентів щодо роботи з географічними об'єктами геоінформаційних баз та визначення їх географічних та геометричних характеристик.

Після виконання завдання студенти повинні:

- *знати* шляхи визначення та набори основних характеристик географічних об'єктів ГІС та деякі основні операції геоаналізу;
- *вміти* різними способами отримувати необхідні характеристики географічних об'єктів, визначати, зокрема й програмно, їх тип та проводити простий геоаналіз.

Робота спирається на знання й уміння, отримані при вивченні "Основ СУБД"

За завданням викладача студенту необхідно розробити шар географічних об'єктів, визначити за допомогою засобів системи та програмно географічні й геометричні характеристики цих об'єктів, провести простий геоаналіз за типами об'єктів: знайти мінімальну та максимальну довготу та широту точкових об'єктів, сумарну довжину об'єктів типу лінія (ламана), сумарну площу та периметр полігонів,

кількість об'єктів типу мітка.

Методичні вказівки

Розробка шару географічних об'єктів. Відповідно до завдання необхідно засобами системи розробити шар географічних об'єктів, нанести ці географічні об'єкти на карту та заповнити деякі атрибутивні дані. Розробка здійснюється в ручному режимі і для таких типів даних як ламана, полігон та мітка використовуються відповідні інструменти системи (рис. 1).

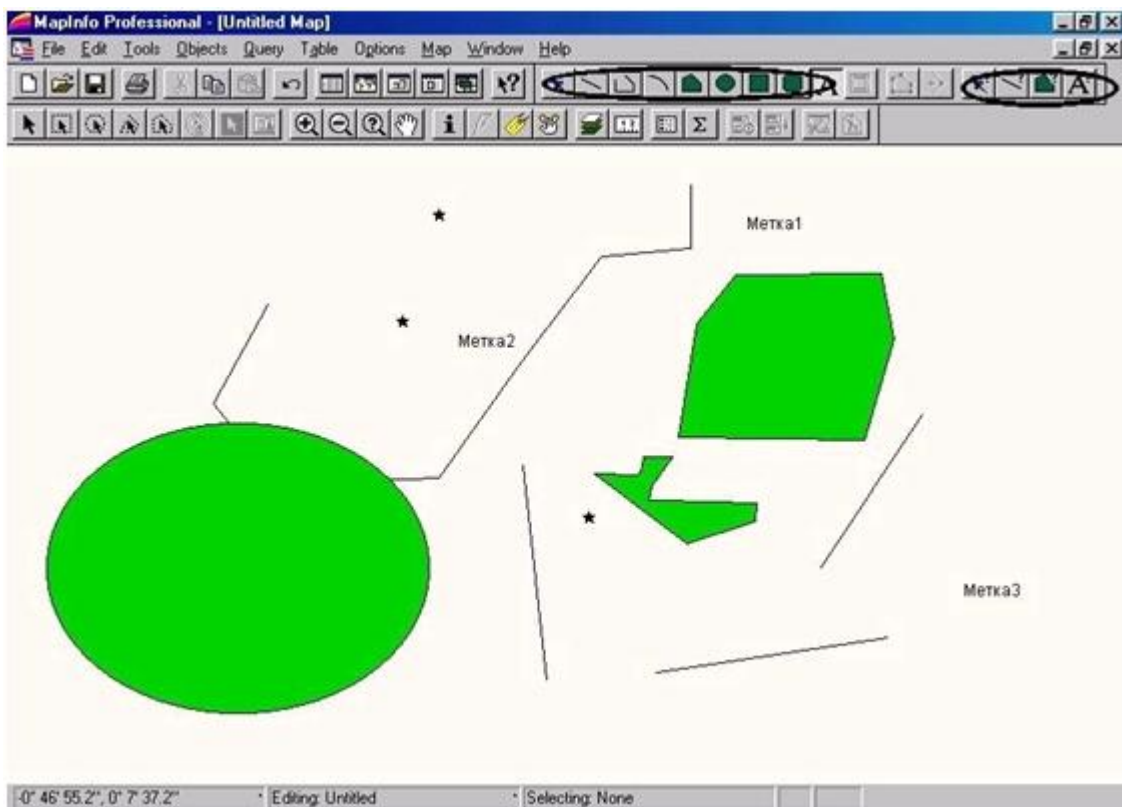


Рис. 1. Занесення географічних даних

При цьому, атрибутивна частина даних має містити такі поля:

- найменування географічного об'єкту (неповторне);
- широта центроїду об'єкта;
- довгота центроїду об'єкта;
- довжина об'єкту;
- периметр об'єкту;
- площа об'єкту.

Слід зауважити наступне, зазвичай пошарове представлення даних в ГІС вимагає розробку шарів з урахуванням типів та необхідних атрибутів географічних об'єктів. Розробка шару географічних об'єктів для робочої

ГІС в якому знаходяться дані різних типів не є раціональною.

Визначення характеристик засобами системи. Спочатку, після нанесення на карту географічних об'єктів необхідно заповнити атрибутивні поля даних. Для заповнення числових атрибутивних полів скористуємось можливостями та засобами що надані системою. Для цього виконаємо команду меню “Таблиця > Замінити стовбець”. При цьому з'явиться діалогове вікно (рис. 2) яке дозволяє визначити який стовбець і як необхідно змінити.

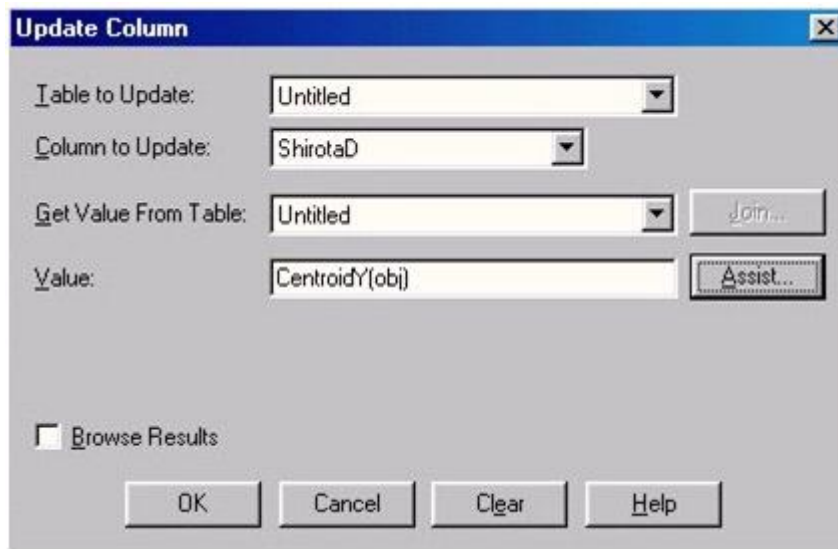


Рис.2. Діалогове вікно для зміни значень стовбців

На рисунку 2 надано приклад зміни поля широти центроїду об'єкта. За допомогою кнопки “Assist...” можна у відповідному вікні (рис. 3) конструювати за допомогою функцій, операторів системи та даних вирази що повертають необхідні дані.

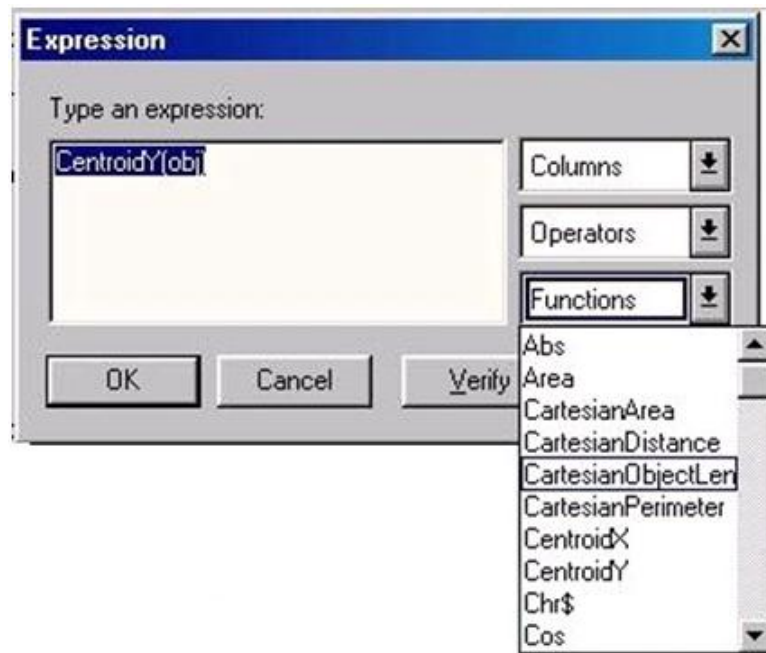


Рис.3. Діалогове вікно конструктора виразів стосовно синтаксису відповідних функцій

Застосовуючи відповідні вирази до стовбців даних належним чином необхідно заповнити атрибутивні дані. Відповідний приклад наведено на рисунку 4.

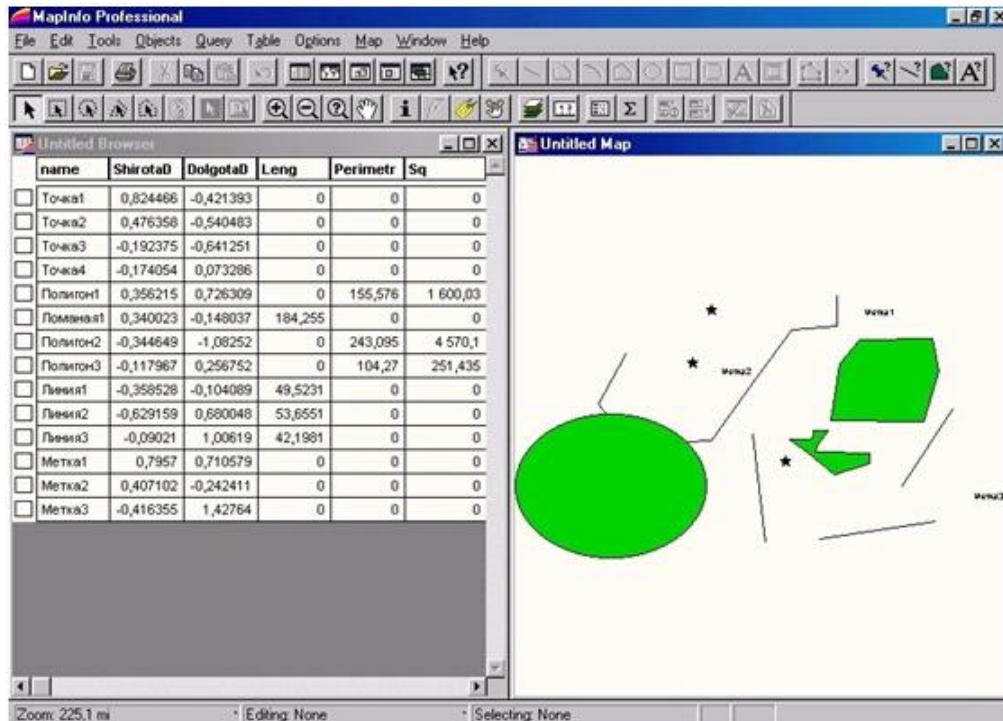


Рис.4. Приклад заповнення атрибутивних даних

Як бачимо на рисунку, багато полів атрибутивних даних мають нульове значення. Дійсно, важко очікувати від точкового об'єкта в якості

його характеристики довжину, площу, периметр, від лінії площу чи периметр, від полігона довжину. Цей приклад ще раз демонструє необхідність раціональної побудови шарів географічних даних ГІС.

Система автоматично розпізнає тип об'єкту та відповідним чином надає значення різних його характеристик. Теж саме можна спостерігати, використовуючи можливості системи щодо отримання інформації по об'єктах. Розглянемо деякі найбільш інформативні з них:

1) Отримання географічної та геометричної інформації по об'єкту. Цим інструментом можна скористатись з допомогою контекстного меню вікна карти (команда "Get Info", клавіша F7). Якщо на карті виділений якийсь об'єкт, по ньому можна отримати геоінформацію. Ця інформація надається системою в діалоговому вікні. На рисунку 5 представлені приклади такої інформації для різних типів об'єктів.

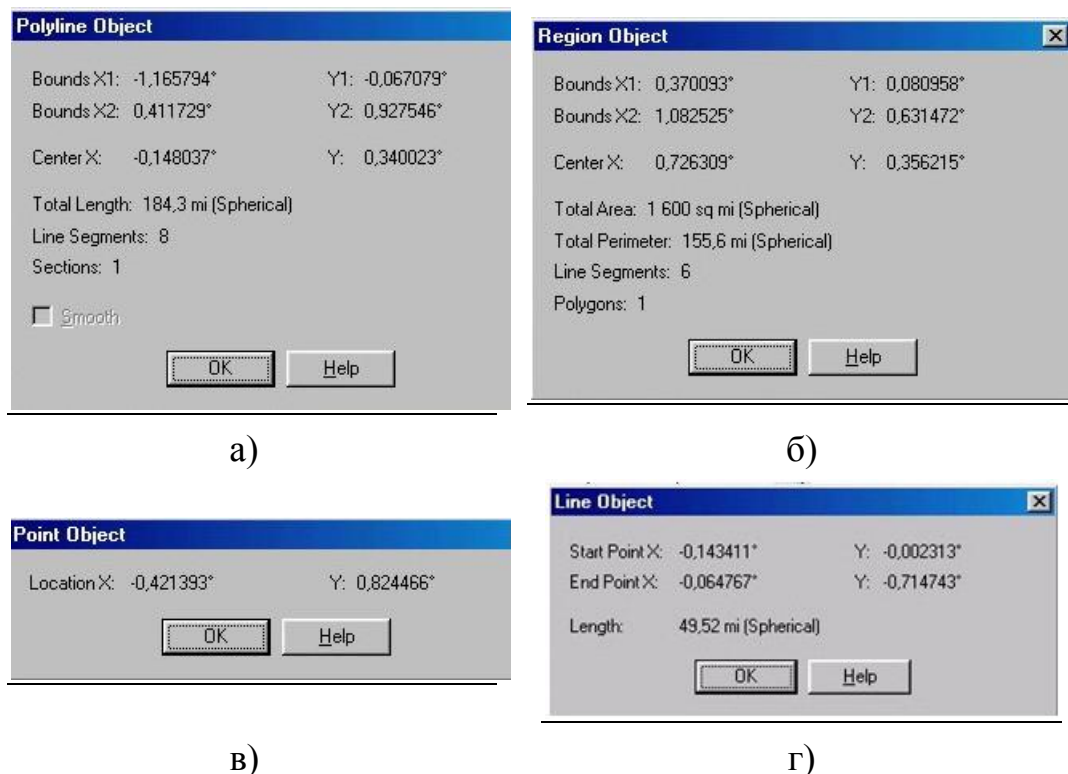


Рис. 5. Приклади геоінформації для об'єктів типу: а – ламана; б – полігон; в – точка; г – лінія

В цьому разі система також надає інформацію відповідно до типу об'єкта.

2) Отримання статистичної інформації по об'єкту. Система дозволяє отримувати різноманітну статистичну інформацію по атрибутивним даним як для одного, для усіх, так і для груп обраних об'єктів. На малюнку 6 представлено вікно статистичних даних по об'єктах (рис. 4). При цьому використовувався підпункт меню "Обчислення статистик" меню "Запити".

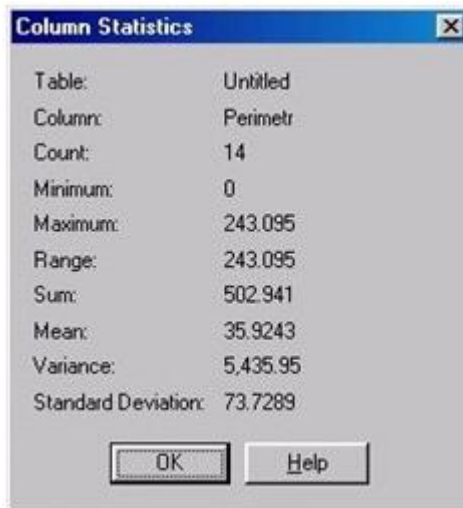


Рис.6. Вікно статистичних даних

Слід зауважити, що використовуючи цей інструмент можна отримати статистичні характеристики лише для одного обраного поля атрибутивних даних. Для отримання меншого набору статистичних характеристик, але по всіх атрибутивних полях відповідного типу, можна скористатись системним вікном статистики (підменю “Показати вікно статистик” меню “Опції”). При цьому, інформація у вікні з’являється лише після обрання того чи іншого об’єкту або групи об’єктів, використовуючи відповідні інструменти системи (рис. 7).

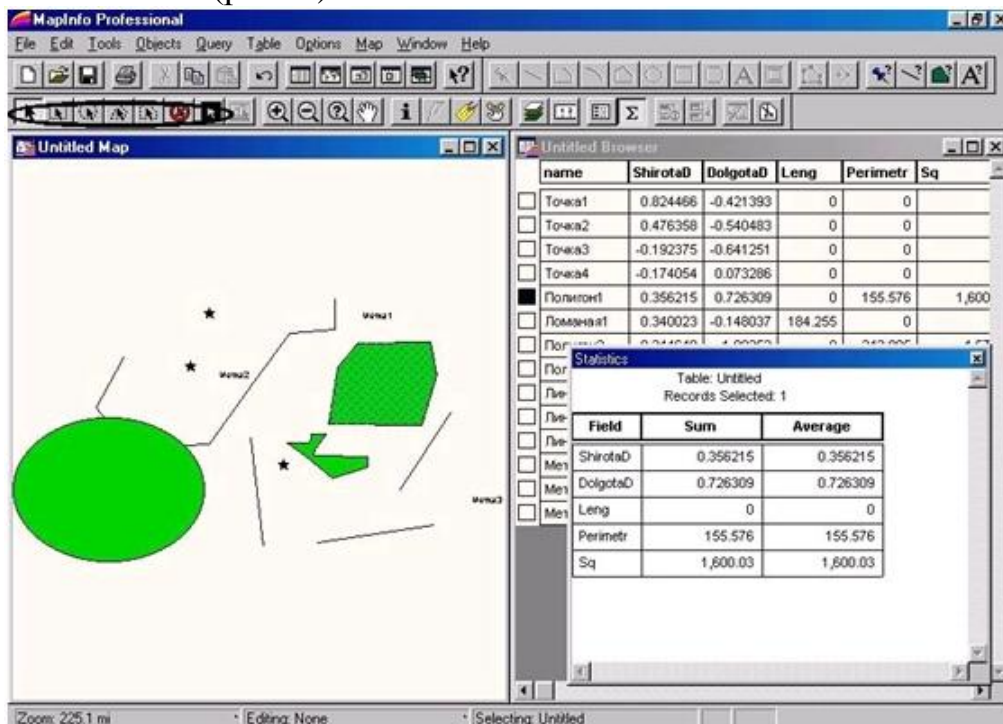


Рис.7. Системне вікно статистик

Важливо зауважити, що для отримання правильних статистичних характеристик для груп об’єктів слід включати їх в групу лише в разі

наявності цих характеристик відповідно до їх типу. Покажемо це на прикладі. На рисунку 6 надано вікно статистик для поля “Периметр” по усіх об’єктах шару, тобто для точок, ліній, поліліній та міток, які не мають такої характеристики. На рисунку 8 наведено вікно статистичних характеристик для того ж поля, але тільки для об’єктів, що мають значення цього поля тобто відповідну геометричну характеристику – полігонів. Можна порівняти зміни значень відповідних статистичних характеристик.

Всі інструменти системи, які призначені для отримання різноманітних статистичних характеристик об’єктів враховують вибір користувача та надають можливість отримувати статистичні характеристики за цим вибором.

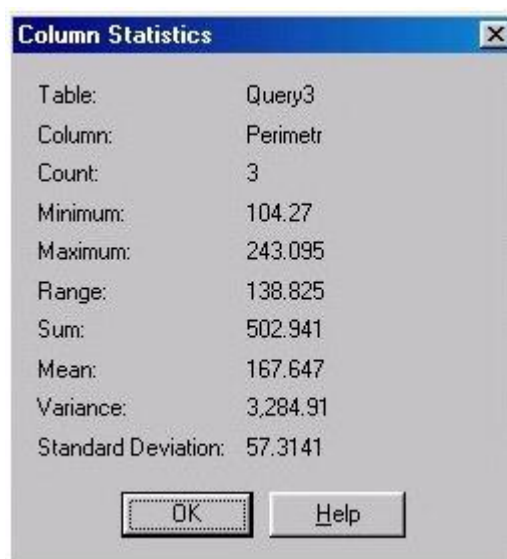


Рис.8. Вікно статистичних даних полігонів

Після створення згідно з завданням геоінформаційного шару, для виконання другої частини роботи необхідно заповнити атрибутивні дані й таблиці 1 та 2 користуючись засобами системи.

Таблиця 1 – Географічні та геометричні дані об’єктів

№ пп	Назва об’єкту	Тип об’єкту	Широта центроїду	Довгота центроїду	Довжина, м	Периметр, м	Площа, км ²

Таблиця 2 – Статистичні дані по типах об’єктів

№ пп	Тип об'єкту	Кількі – сть	Середня широта	Середня довгота	Сумарна (середня) довжина, м	Сумарний (середній) периметр, м	Сумарна (середня) площа, км ²

Програмне визначення характеристик географічних об'єктів.
Завдання третьої частини роботи полягає в розробці програмного додатку який, використовуючи результати виконання першої частини лабораторної роботи має виконувати наступне:

- 1) заповнювати відповідні атрибутивні поля розробленого шару об'єктів;
- 2) в разі вибору того чи іншого об'єкту карти, відповідно до його типу надавати (чи в діалоговому вікні, чи в системному вікні повідомлень) разом географічні, геометричні та статистичні характеристики цього об'єкту;
- 3) автоматично створювати та записувати в текстовий файл таблиці 2.1, 2.2

Нижче надано опис деяких функцій та інструкції, які можуть знадобитись при виконанні третьої частини лабораторної роботи.

Area() function

Призначення: Повертає площу географічного об'єкта.

Синтаксис: Area (*obj_expr* , *unit_name*)

obj_expr - об'єктний вираз;

unit_name - строкове вираження, що представляє одиниці виміру площі (наприклад "sq km").

Значення, що повертає: Float.

Опис: Area() функція повертає площу географічного об'єкта, зазначеного *obj_expr*.

Функція повертає значення площі в одиницях, зазначених *unit_name* па- параметром; наприклад, щоб одержати область в акрах, визначите "acre" як *unit_name* параметр. Див. інструкцію Set Area Units для списку доступних назв одиниць.

Тільки регіони, еліпси, прямокутники, і округлені прямокутники мають площу. За визначенням, Area() крапки, дуги, тексту, лінії, або об'єкта ламаної лінії - нуль. Функція Area() повертве апроксимовані результати для скруглених прямокутників.

Centroid() function

Призначення: Повертає центроїд (середня точка) об'єкта.

Синтаксис: Centroid(*obj_expr*)

obj_expr - object вираз;

Значення, що повертає: Point object

Опис: Centroid() функція повертає точковий об'єкт, що розташований у центроїді зазначеного об'єкту *obj_expr*.

Зверніть увагу: В MapInfo, центроїд області не є його центром маси. Замість цього, центроїд представляє положення, використовуване для автоматичного маркування, геокодування, і розміщення тематичного кола й гістограм.

Якщо *obj_expr* параметр представляє точковий об'єкт, Centroid() повертає позицію точки.

Якщо *obj_expr* параметр представляє об'єкт лінії, Centroid() повертає крапку на посередині між кінцями лінії.

Якщо *obj_expr* параметр представляє об'єкт ламаної лінії, Centroid() повертає крапку, розташовану всередині середнього сегменту ламаної лінії.

Якщо *obj_expr* параметр представляє будь-який інший тип об'єкту, Centroid() повертає крапку, розташовану в дійсному центроїді первісного об'єкта. Для прямокутника, дуги, тексту, і об'єктів еліпса, позиція центроїда - посередині між верхнім і нижнім розмірами об'єкта, і між лівим і правим розмірами. Для об'єктів області, однак, позиція центроїда - завжди "на" розглянутому об'єкті, і тому не може бути розташована посередині між границями об'єкта.

CentroidX(), CentroidY() function

Призначення: Повертають відповідно довготу та широту центроїда об'єкта.

Синтаксис: CentroidX(*obj_expr*), CentroidY(*obj_expr*):

obj_expr - object вираз. Значення, що повертає: Float

Опис: CentroidY() (CentroidX()) функція повертає Y(X)-координату (широта(довгота)) центроїда зазначеного об'єкту.

Perimeter() function

Призначення: Повертає периметр графічного об'єкта.

Синтаксис: Perimeter (*obj_expr* , *unit_name*) *obj_expr* - об'єктний вираз; *unit_name* - рядок, що представляє назву одиниць виміру відстані (наприклад "km").

Значення, що повертає: Float.

Опис: Perimeter() функція обчислює периметр об'єкта *obj_expr*.

Perimeter() визначена для наступних об'єктів: еліпси, прямокутники, скруглені прямокутники, і багатокутники. Інші типи об'єктів мають периметр - нуль.

Perimeter() повертає вимір довжини в одиницях, зазначених *unit_name* параметром; наприклад, щоб одержати довжину в милях, визначите "mi" як *unit_name* параметр. Див. інструкцію Set Distance Units для списку наявних назв одиниць.

Perimeter() апроксимує результати для скруглених прямокутників. Можна також використовувати Perimeter() функцію в SQL інструкції Select.

Distance() function

Призначення: Повертає відстань між двома точками.

Синтаксис: Distance (*x1* , *y1* , *x2* , *y2* , *unit_name*)

x1 і *x2* - *x* - координати (напр. довгота);

y1 і *y2* - *y* - координати (напр. широта);

unit_name - рядок, що представляє назву одиниць виміру відстані (наприклад "km").

Значення, що повертає: Float.

Опис: Distance() функція обчислює відстань між двома крапками.

Функція повертає вимір відстані в одиницях, зазначених *unit_name* пара- метром; наприклад, щоб одержати відстань у милях, визначите "mi" як *unit_name* параметр.

Параметри *x*, *y* координат повинні використовувати поточну систему координат MapBasic. За замовчуванням, MapInfo очікує, що координати будуть використовувати долготно-широтну, систему координат.

Якщо поточна система координат - земна система координат, Distance() повертає відстань великого кола між двома крапками. Відстань великого кола - сама коротка відстань між двома точками на сфері. (Велике коло – це коло, що йде навколо землі, із центром кола в центрі землі; відстань великого кола між двома крапками - відстань по великому колу, що з'єднує дві точки.)

Якщо поточна система координат - неземна система координат, Distance() повертає Декартову (картезіанську) відстань.

ObjectLen() function

Призначення: Повертає географічну довжину об'єкта лінії або полілінії.

Синтаксис: ObjectLen(*expr* , *unit_name*)

expr – об'єктне вираження;

unit_name - рядок, що представляє назву одиниць виміру відстані

(наприклад "mi" для миль).

Значення, що повертає: Float.

Опис: ObjectLen() функція повертає довжину об'єкта. Тільки лінія й об'єкти ламаної лінії мають значення довжини більше чим нуль; щоб вимірювати окружність прямокутника, еліпс, або область, використовують функцію Perimeter().

ObjectLen() функція повертає довжину в одиницях, зазначених *unit_name* параметром; наприклад, щоб одержати довжину в милях, визначите "mi" як *unit_name* параметр.

MapperInfo() function

Призначення: Повертає інформацію про координати або відстань для вікна *Map*.

Синтаксис: MapperInfo(*window_id* , *attribute*)

window_id - Integer ідентифікатор вікна;

attribute - Integer код, призначений для вказівки необхідної інформації.

Значення, що повертає: Float, Logical, або String, залежить від *параметра attribute*

Опис: MapperInfo() функція повертає інформацію про вікно *Map*.

Window_id параметр визначає вікно *Map* для якого робиться запит. Щоб одержати ідентифікатор вікна, викличте FrontWindow() функцію негайно після відкриття вікна, або викличте WindowID() у будь-який час після створення вікна.

Є кілька числових атрибутів, які MapperInfo() може повертати про будь-яке задане вікно *Map*. Параметр *attribute* повідомляє MapperInfo() функції яка статистика вікна *Map* необхідна. Параметр *attribute* повинен мати одне зі значень із наступної таблиці 2.3; коди визначені в MapBasic .DEF.

Таблиця 2.3 – Значення параметру *Attribute* та відповідне значення функції MapperInfo()

Attribute	MapperInfo()
MAPPER_INFO_AREAUNITS	Рядок, що представляє скорочену назву одиниць площі карти, наприклад "sq mi" для квадратних миль.
MAPPER_INFO_CENTERX	X-координата центра вікна <i>Map</i>
MAPPER_INFO_CENTERY	Y-координата центра вікна <i>Map</i>

MAPPER_INFO_COORDSYS_CLAUSE	Рядок, що вказує CoordSys вікна
MAPPER_INFO_COORDSYS_NAME	Рядок, що представляє назву CoordSys карти які перераховані в MAPINFOW.PRJ. Повертає порожній рядок, якщо CoordSys не знайдена в MAPINFOW.PRJ
MAPPER_INFO_DISPLAY	Small integer, вказує, який аспект карти відображений в рядку стану. Відповідає Set Map Display. Значення, що повертає, буде одним із цих: MAPPER_INFO_DISPLAY_SCALE MAPPER_INFO_DISPLAY_ZOOM MAPPER_INFO_DISPLAY_POSITION
MAPPER_INFO_DISTUNITS	Рядок, що представляє скорочену назву одиниць відстані карти, наприклад "mi" для милі
MAPPER_INFO_EDIT_LAYER	SmallInt вказує номер доступного для редагування шару в даний час. Значення нуль означає, що косметичний шар доступний для редагування. Значення -1 означає, що ніякий шар не доступний для редагування.
MAPPER_INFO_LAYERS	Повертає число рівнів у вікні Map як SmallInt (крім косметичного шару)
MAPPER_INFO_MAXX	Сама більша x-координата, показана у вікні
MAPPER_INFO_MAXY	Сама більша y-координата, показана у вікні
MAPPER_INFO_MINX	Сама маленька x-координата, показана у вікні
MAPPER_INFO_MINY	Сама маленька y-координата, показана у вікні
MAPPER_INFO_NUM_THEMATIC	SmallInt, вказує число тематичних рівнів у цьому вікні Map
MAPPER_INFO_SCALE	Поточний масштаб вікна Map, визначений як відношення одиниць відстані карти (наприклад милі) до паперових одиниць (наприклад дюйми)

	відображеної у вікні карти. Повертає значення в поточних одиницях відстані MapBasica
MAPPER_INFO_SCROLLBARS	Логічне значення, що вказує, чи відображує вікно Map scrollbars
MAPPER_INFO_XYUNITS	Рядок, що представляє скорочену назву одиниць координат карти, наприклад "градус".
MAPPER_INFO_ZOOM	Поточне значення зміни розміру зображення вікна Map (zoom) (тобто відстань схід-захід у цей час відображуване у вікні Map), зазначене в поточних одиницях відстані MapBasica

Коли викликається MapperInfo() щоб одержати значення координат (наприклад, визначаючи MAPPER_INFO_CENTERX як атрибут), повернуте значення представляє координату в поточній системі координат MapBasic, що може відрізнятися від системи координат вікна Map. Використовуйте інструкцію Set CoordSys, щоб визначити іншу систему координат.

LayerInfo() function

Призначення: Повертає інформацію про шар у вікні Map.

Синтаксис: LayerInfo(*map_window_id,layer_number,attribute*)

map_window_id - ідентифікатор вікна Map;

layer_number - номер шару в поточному вікні Map (наприклад 1 для верхнього шару); щоб визначити число шарів у вікні Map, викличте MapperInfo();

attribute - код, що вказує тип необхідної інформації; див. таблицю нижче.

Значення, що повертає: Повертає значення, що, залежить від параметра *attribute*.

Обмеження: Багато з параметрів які використовуються LayerInfo() звертаються тільки до звичайних шарів карти (на противагу косметичним, тематичним шарам карти, і шарам, що представляють таблиці растрових зображень).

Опис: LayerInfo() функція повертає інформацію про один шар в існуючому вікні Map. *Layer_number* повинен вказувати шар (0 - косметичний шар, 1 – самій верхній (topmost) шар таблиці, і так далі). Параметр *attribute* повинен мати один з кодів з наступної таблиці 2.4; коди визначені в MapBasic.DEF.

Таблиця 2.4 – Значення параметру Attribute та відповідне значення функції LayerInfo()

Attribute	MapperInfo()
LAYER_INFO_EDITABLE	Логічне значення; TRUE, якщо шар можна редагувати
LAYER_INFO_SELECTABLE	Логічне значення; TRUE, якщо в шарі можна обирати об'єкти
LAYER_INFO_NAME	Рядок, що вказує ім'я таблиці, пов'язаної із цим шаром карти. Якщо зазначений шар – косметичний, рядок буде ім'ям таблиці типу "Cosmetic1"; це ім'я таблиці може використовуватись з іншими інструкціями (наприклад. Select)
LAYER_INFO_PATH	Рядок, що представляє повний шлях до зв'язаної із шаром карти таблиці
LAYER_INFO_ZOOM_LAYERED	Логічне; TRUE, якщо допускається змінювати розміри зображення шару.
LAYER_INFO_ZOOM_MIN	Float значення, вказує мінімум масштабу зображення (у поточних одиницях відстані MapBasic) у якому шар відображається
LAYER_INFO_ZOOM_MAX	Float значення, укажує максимум масштабу зображення, у якому шар відображається
LAYER_INFO_COSMETIC	Логічний; TRUE, якщо це - косметичний шар
LAYER_INFO_DISPLAY	SmallInt, вказує як і де цей шар відображений; Повертає значення, що, буде одним із цих: LAYER_INFO_DISPLAY_OFF (Шар не відображений); LAYER_INFO_DISPLAY_GRAPHIC (Об'єкти в цьому шарі з'являються в їх "заданому за замовчуванням" стилі - стиль, збережений у таблиці); LAYER_INFO_DISPLAY_GLOBAL (Об'єкти в цьому шарі відображені зі "скасуванням стилю" зазначеному в Layer Control);

	LAYER_INFO_DISPLAY_VALUE (Об'єкти в цьому шарі з'являються як тематичні)
LAYER_INFO_OVR_LINE	Pen стиль, використовуваний для подання лінійних об'єктів
LAYER_INFO_OVR_PEN	Pen стиль, використовуваний щоб показувати границі заповнених об'єктів
LAYER_INFO_OVR_BRUSH	Brush стиль, використовуваний для показу заповнених об'єктів
LAYER_INFO_OVR_SYMBOL	Symbol стиль, використовуваний для показу об'єктів типу точка
LAYER_INFO_OVR_FONT	Font стиль, використовуваний для показу текстових об'єктів
LAYER_INFO_LBL_CURFONT	Для програм, відкомпільованих MapBasic 4.0, цей запит завжди повертає FALSE. Для програм, відкомпільованих MapBasic 3.x, цей запит повертає наступні значення: TRUE, якщо шар установлений, щоб використати поточний шрифт, або FALSE, якщо шар установлений, щоб використати замовлений шрифт (див. LAYER_INFO_LBL_FONT)
LAYER_INFO_LBL_FONT	Font стиль, використовуваний у мітках
LAYER_INFO_LBL_EXPR	String значення: вираження, використовуване в мітках
LAYER_INFO_LBL_LT	Smallint значення, що вказує на якій лінії, якщо такі взагалі є, підключати мітку в первісне положення після того, як перемістили мітку. Значення, що повертається, буде відповідати одному із цих: LAYER_INFO_LBL_LT_NONE (немає лінії) LAYER_INFO_LBL_LT_SIMPLE (проста лінія) LAYER_INFO_LBL_LT_ARROW (лінія з курсором arrowhead)
LAYER_INFO_LBL_PARALLEL	Логічне значення TRUE, якщо шар установлений для паралельних міток

LAYER_INFO_LBL_POS	<p>Smallint значення, указує позицію мітки. Значення, що повертається, буде відповідати одному із цих (T=Top, B=Bottom, C=Center, R=Right, L=Left):</p> <p>LAYER_INFO_LBL_POS_TL LAYER_INFO_LBL_POS_TC LAYER_INFO_LBL_POS_TR LAYER_INFO_LBL_POS_CL LAYER_INFO_LBL_POS_CC LAYER_INFO_LBL_POS_CR LAYER_INFO_LBL_POS_BL LAYER_INFO_LBL_POS_BC LAYER_INFO_LBL_POS_BR</p>
LAYER_INFO_LBL_VISIBILITY	<p>Smallint значення, указує, чи є мітки видимими. Значення, що повертається, буде одним із цих:</p> <p>LAYER_INFO_LBL_VIS_ON (мітки завжди видимі) LAYER_INFO_LBL_VIS_OFF (мітки завжди невидимі) LAYER_INFO_LBL_VIS_ZOOM (мітки видимий, після зміни діапазону)</p>
LAYER_INFO_LBL_ZOOM_MIN	Float значення, указує мінімум відстані для відображення міток цього шару
LAYER_INFO_LBL_ZOOM_MAX	Float значення, указує максимум відстані для відображення міток цього шару
LAYER_INFO_LBL_AUTODISPLAY	Логічне значення: TRUE, якщо цей шар установлений, щоб відобразити мітки автоматично
LAYER_INFO_LBL_OVERLAP	Логічне значення; TRUE якщо перекриття міток дозволяється
LAYER_INFO_LBL_DUPLICATES	Логічне значення; TRUE, якщо подвійні мітки дозволяються
LAYER_INFO_LBL_OFFSET	Smallint значення від 0 до 50, вказує, як далеко мітки зміщені від центру об'єкта. Значення зсуву представляє відстань, у крапках
LAYER_INFO_LBL_MAX	Integer значення, вказує максимальну кількість міток, дозволених для цього

	шару. Якщо максимум не був встановлений - повертає 2,147,483,647
LAYER_INFO_LBL_PARTIALSEGMENTS	Логічне значення; TRUE, якщо перемикач Label Partial Segments встановлений для цього шару
LAYER_INFO_ARROWS	Логічне значення; TRUE, якщо шар відображає стрілки напрямків на лінійних об'єктах (цілях).
LAYER_INFO_NODES	Логічне значення; TRUE, якщо шар відображає вузли об'єктів
LAYER_INFO_CENTROIDS	Логічне значення; TRUE, якщо шар показує центроїди об'єктів
LAYER_INFO_TYPE	SmallInt значення, указує тип файлу цього шару: LAYER_INFO_TYPE_NORMAL для нормального шару; LAYER_INFO_TYPE_COSMETIC косметичного шару; LAYER_INFO_TYPE_IMAGE для шару растрового зображення; LAYER_INFO_TYPE_THEMATIC тематичного шару

ObjectGeography() function

Призначення: Повертає координатну або кутову інформацію, що описує графічний об'єкт.

Синтаксис: ObjectGeography(*object* , *attribute*)

object - об'єкт вираз;

attribute - Integer код, що визначає тип повертає інформації.

Значення, що повертає: Float.

Опис: *Attribute* параметр управляє типом інформації, що повертається.

Таблиця 2.5 надає різні коди, які можна використати як параметр атрибута; коди визначені в MAPBASIC.DEF.

Деякі атрибути застосовуються тільки для деяких типів об'єктів. Наприклад, об'єкти дуги - єдині об'єкти з початковими або кінцевими кутовими атрибутами і текстові об'єкти - єдині об'єкти з кутовим атрибутом тексту.

Таблиця 2.5 – Значення параметру Attribute та відповідне значення

функції ObjectGeography()

Attribute	MapperInfo()
OBJ_GEO_MINX	Мінімальна x координата мінімального обмежуючого прямокутника (МОП) об'єкта, якщо об'єкт не лінія; якщо об'єкт - лінія, повертає те ж саме значення що й OBJ_GEO_LINEBEGX.
OBJ_GEO_MINY	Мінімальна y координата МОП об'єкта. Для ліній, повертає OBJ_GEO_LINEBEGY значення
OBJ_GEO_MAXX	Максимальна x координата МОП об'єкта. Не застосовується до об'єктів Point. Для ліній, повертає значення OBJ_GEO_LINEENDX
OBJ_GEO_MAXY	Максимальна y координата МОП об'єкта. Не застосовується до об'єктів Point. Для ліній, повертає значення OBJ_GEO_LINEENDY
OBJ_GEO_ARCBEGANGLE	Початковий кут об'єкта Arc.
OBJ_GEO_ARCENDANGLE	Кінцевий кут об'єкта Arc
OBJ_GEO_LINEBEGX	x координата стартового вузла об'єкта Line
OBJ_GEO_LINEBEGY	y координата стартового вузла об'єкта Line
OBJ_GEO_LINEENDX	x координата кінцевого вузла об'єкта Line
OBJ_GEO_LINEENDY	y координата кінцевого вузла об'єкта Line
OBJ_GEO_POINTX	x координата об'єкта Point.
OBJ_GEO_POINTY	y координата об'єкта Point.
OBJ_GEO_ROUNDADIUS	Діаметр кола, що визначає кут скруглення скругленого прямокутника, виражений у термінах

	координатних одиниць (наприклад градуси).
OBJ_GEO_TEXTLINEX	x координата кінця лінії мітки об'єкта Text
OBJ_GEO_TEXTLINEY	y координата кінця лінії мітки об'єкта Text
OBJ_GEO_TEXTANGLE	Кут оберту об'єкта Text

MBR() function

Призначення: Повертає об'єкт прямокутник, що представляє мінімально обмежуючий прямокутник іншого об'єкта.

Синтаксис: MBR(*obj_expr*)

obj_expr – об'єктний вираз;

Значення, що повертає: Object (прямокутник)

Опис: MBR() функція обчислює мінімально обмежуючий прямокутник (або МОП) який охоплює зазначений об'єкт *obj_expr*.

Мінімально обмежуючий прямокутник є самим маленьким прямокутником, що є достатнім, щоб охопити специфічний об'єкт. МОП точкового об'єкта має нульову ширину й нульову висоту.

ObjectNodeX() function

Призначення: Повертає x-координату певного вузла в об'єкті ламаної лінії або області.

Синтаксис: ObjectNodeX(*object* , *polygon_num* , *node_num*)

object - Object вираз;

polygon_num - позитивне Integer значення, що ідентифікує полігон або секцію, до якої робиться запит;

node_num - позитивне Integer значення, що ідентифікує вузол.

Значення, що повертає: Float.

Опис: ObjectNodeX() функція повертає x-значення певного вузла об'єкта ламаної лінії або області.

Polygon_num параметр повинен мати значення 1 або більше. Він визначає необхідний багатокутник (якщо запитується область) або секцію (якщо запитується ламана лінія). Викличте ObjectInfo() функцію щоб визначити число багатокутників або секцій в об'єкті.

Node_num параметр повинен мати значення 1 або більше; він повідомляє MapBasic, опитуваний вузол об'єкта. Ви можете використати ObjectInfo() функцію, щоб визначити число вузлів в об'єкті.

Об'єкт tNodeX() функція поверне значення в системі координат, використовуваний в цей час MapBasic; за замовчуванням MapBasic

використовує долготно-широтну систему координат.

ObjectNodeY() function

Призначення: Повертає у-координату певного вузла в об'єкті ламаної лінії або області.

Синтаксис: ObjectNodeY(*object*, *polygon_num*, *node_num*)

object - Об'єкт вираз;

polygon_num - позитивне Integer значення, що ідентифікує полігон або секцію, до якої робиться запит;

node_num - позитивне Integer значення, що ідентифікує вузол.

Значення, що повертає: Float.

Опис: ObjectNodeY() функція повертає у-значення певного вузла об'єкта ламаної лінії або області. Див. опис ObjectNodeX() функції для додаткової інформації.

Set Area Units statement

Призначення: Встановлює задану за замовчуванням одиницю площі в MapBasic.

Синтаксис: Set Area Units *area_name*

area_name - рядок, що представляє назву одиниці площі (наприклад "acre").

Опис: Інструкція Set Area Units установлює задану за замовчуванням одиницю виміру площі MapInfo. Вона визначає одиницю площі, використовувану в межах діалогу Select SQL MapInfo. За замовчуванням, MapBasic використовує квадратні милі; ця одиниця залишається, якщо інструкція Set Area Units не викликала.

Area_name параметр повинен мати одне зі строкових значень, перерахованих у таблиці 2.6:

Таблиця 2.6 – Значення параметру *Area_name*

Area_name	Одиниця
"acre"	Акри
"hectare"	Гектари
"perch"	Перчи
"rood"	чверті акра

"sq ch"	кв. ланцюга
"sq cm"	кв. сантиметри
"sq ft"	кв. фути
"sq in"	кв. дюйми
"sq km"	кв. кілометри
"sq li"	кв. линки(links)
"sq m"	кв. метри
"sq mi"	кв. милі
"sq mm"	кв. міліметри
"sq rd"	кв. пологи
"sq survey ft"	кв. топографічні фути
"sq yd"	кв. ярд

Set Distance Units statement

Призначення: Установлює одиницю відстані, використовувану для наступних географічних операцій, типу Create Object.

Синтаксис: Set Distance Units *unit_name*

unit_name - назва одиниці відстані (наприклад "m" для метрів).

Опис: Інструкція Set Distance Units установлює лінійну одиницю виміру MapBasic. За замовчуванням, MapBasic використовує одиницю відстані "mi" (мили); ця одиниця відстані залишається, якщо інструкція Set Distance Units не запускалаась.

Деякі інструкції MapBasic беруть параметри, що представляють відстані. Наприклад, пропозиція Width інструкції Create Object може або не може визначати одиницю відстані. Якщо пропозиція Width не визначає модуль відстані, Create Object використає поточні одиниці відстані (або милі або визначені останньою інструкцією Set Distance Units). *Unit_name* параметр повинен мати одне зі значень із таблиці 2.7.

Таблиця 2.7 – Значення параметру *Unit_name*

Unit_name	Назва
"ch"	Ланцюги (chain19.8 м)

"cm"	Сантиметри
"ft"	(інтернаціональний фут = 30.48 cm) Фути
"in"	Дюйми
"km"	Кілометри
"li"	Лінки (links)
"m"	Метри
"mi"	Милі
"mm"	Міліметри
"nmi"	Навігаційні (морські) милі (1852 м)
"rd"	Rods (≈5м)
"survey ft"	U.S. топографічні фути (исп. для координат рівнів штатів 1927р.; один U.S. топфут = 12/39.37 метрів, або 30.48006 cm)
"yd"	Ярди

ObjectInfo() function

Призначення: Повертає Pen, Brush, або інші значення, що описують графічний об'єкт.

Синтаксис: ObjectInfo(*object*, *attribute*)

object - Object вираз;

attribute - Integer код визначає тип інформації, що повинен бути повернутий.

Значення, що повертає: SmallInt, Integer, String, Float, Pen, Brush, Symbol, або Font, залежно від параметра *attribute*.

Описання: ObjectInfo() функція повертає загальну інформацію про один аспект графічного об'єкта. Перший параметр повинен бути об'єктним (наприклад ім'я об'єктної змінної, або табличним вираженням у формі tablename.obj).

Кожний об'єкт має кілька атрибутів. Наприклад, кожний об'єкт має атрибут "type", що ідентифікує об'єкт як крапку, лінію, або область, і т.д.

Більшість типів об'єктів має Pen і/або Brush атрибути, які визначають вид об'єкта. ObjectInfo() функція повертає один атрибут певного об'єкта. Який атрибут повертається, залежить від значення, використаного в параметрі *attribute*. Таким чином, якщо необхідно з'ясувати кілька аспектів інформації про об'єкт, треба викликати ObjectInfo() неодноразово, з різними значеннями *attribute* у кожному запиті.

Таблиця 2.8 надає різні значення параметру *attribute*, і відповідні значення, що повертає ObjectInfo().

Attribute	MapperInfo()
OBJ_INFO_TYPE	SmallInt, що представляє тип об'єкта; повертає одне зі значень представлених у таблиці 2.9 (наприклад OBJ_TYPE_LINE)
OBJ_INFO_PEN	Повертає Pen стиль; цей запит має силу тільки для наступних типів об'єктів: Дуга, Еліпс, Лінія, Ламана лінія, Рамка, Області, Прямокутник, Скруглений прямокутник
OBJ_INFO_BRUSH	Повертає стиль Brush; цей запит дійсний для наступних об'єктів: Еліпс, Рамка, Область, Прямокутник, Скруглений прямокутник.
OBJ_INFO_TEXTFONT	Повертає стиль Шрифту; дійсний для об'єктів Text
OBJ_INFO_SYMBOL	Стиль Symbol; цей запит дійсний для об'єктів Point
OBJ_INFO_NPNTS	Integer, що вказує загальну кількість вузлів у ламаній лінії або області.
OBJ_INFO_SMOOTH	Logical, що вказує, чи згладжений зазначений об'єкт Polyline.
OBJ_INFO_FRAMEWIN	Integer, що вказує ідентифікатор вікна, вкладеного в об'єкт Frame (рамка).
OBJ_INFO_FRAMETITLE	String, що вказує заголовок об'єкта Frame
OBJ_INFO_NPOLYGONS	SmallInt, що вказує число полігонів (у випадку області) або секцій (у випадку ламаної лінії) які становлять об'єкт
OBJ_INFO_NPOLYGONS+N	Integer, що вказує число вузлів в n-ному багатокутнику області або n-ній

	секції ламаної лінії. Зверніть увагу: з об'єктами області, MapInfo розраховує стартовий вузол двічі (як вузол початку і як кінцевий вузол). Наприклад, ObjectInfo повертає значення 4 для маючої форму трикутника області
OBJ_INFO_TEXTSTRING	String, що представляє тіло об'єкта Text; якщо об'єкт має множинну лінію тексту, рядок включає впроваджені переводи рядка (Chr\$(10) значення)
OBJ_INFO_TEXTSPACING	Float значення 1, 1.5, або 2, представляє міжрядковий інтервал об'єкта Text
OBJ_INFO_TEXTJUSTIFY	SmallInt, представляє вирівнювання об'єкта Text: 0 = уліво, 1 = по центру, 2 = вправо
OBJ_INFO_TEXTARROW	SmallInt, представляє стиль лінії, пов'язаної з об'єктом Text: 0 = немає лінії, 1 = проста лінія, 2 = лінія зі стрілкою
OBJ_INFO_FILLFRAME	Logical: TRUE якщо об'єкт - рамка, що містить вікно Map, і настроювання "Fill Frame With Map" установлені

Коди в лівому стовпці (наприклад. OBJ_INFO_TYPE) визначені у файлі визначень MapBasic, MAPBASIC.DEF. Програма повинна включити "MAPBASIC.DEF", якщо необхідно визивати ObjectInfo() функцію.

Якщо визначається OBJ_INFO_TYPE як значення *attribute*, ObjectInfo() функція повертає один з об'єктних типів, перерахованих у таблиці 2.9.

Таблиця 2.9 –Значення, що повертає ObjectInfo() за типами об'єктів

Attribute	MapperInfo()
OBJ_TYPE_ARC	Arc об'єкт
OBJ_TYPE_ELLIPSE	Ellipse / circle objects
OBJ_TYPE_LINE	Line об'єкт
OBJ_TYPE_PLINE	Polyline об'єкт

OBJ_TYPE_POINT	Point об'єкт Layout window
OBJ_TYPE_FRAME	Frame об'єкт
OBJ_TYPE_REGION	Region об'єкт
OBJ_TYPE_RECT	Rectangle об'єкт
OBJ_TYPE_ROUNDRECT	Rounded rectangle об'єкт
OBJ_TYPE_TEXT	Text об'єкт

TableInfo() function

Призначення: Повертає інформацію про відкриту таблицю.

Синтаксис: TableInfo(*table_id* , *attribute*)

table_id - String – ім'я таблиці, позитивний Integer – номер таблиці, або 0 (нуль);

attribute - Integer код, що визначає аспект який потрібно повернути.

Значення, що повертає: String, SmallInt, або Logical, зачисит *від attribute*.

Опис: TableInfo() функція повертає одну частину інформації про відкриту таблицю.

Table_id може бути рядком, що представляє ім'я відкритої таблиці. Також, *table_id* може бути номером таблиці. Якщо *table_id* - 0 (нуль), TableInfo() повертає інформацію про останню відкриту або створену таблицю; це дозволяє програмі MapBasic визначати робоче ім'я таблиці у випадках, де інструкція Open Table не включала пропозицію As. Якщо немає ніяких відкритих таблиць, або остання таблиця вже закрита, TableInfo() генерує помилку.

Параметр *attribute* може мати будь-яке значення з таблиці 2.10. Коди в лівому стовпці (наприклад - TAB_INFO_NAME) визначені в MAPBASIC .DEF.

Таблиця 2.10 – Параметр *attribute*, і відповідні значення TableInfo() Attribute TableInfo()

Attribute	MapperInfo()
TAB_INFO_COORDSYS_CLAUSE	String результат, вказує пропозицію CoordSys таблиці, типу "CoordSys Earth Projection 1, 0". Повертає порожній рядок, якщо

	таблиця - не mappable.
TAB_INFO_EDITED	Logical результат; TRUE, якщо таблиця не зберегла редагування
TAB_INFO_COORDSYS_MINX, TAB_INFO_COORDSYS_MINY, TAB_INFO_COORDSYS_MAXX, TAB_INFO_COORDSYS_MAXY	Float результати, що вказують мінімум або максимум x або y координати карти які таблиця може зберігати; якщо таблиця - не mappable, повертає нуль
TAB_INFO_COORDSYS_NAME	String результат, представляє назву CoordSys яка перерахована в MAPINFOW.PRJ (але без додаткового "\r ... " суфікса, що з'являється в MAPINFOW.PRJ). Повертає порожній рядок, якщо таблиця - не mappable, або якщо CoordSys не знайдений в MAPINFOW.PRJ.
TAB_INFO_FASTEDIT	Logical результат; TRUE, якщо таблиця має FastEdit включений режим, FALSE інакше.
TAB_INFO_MAPPABLE	Logical результат; TRUE, якщо таблиця -mappable.
TAB_INFO_MAPPABLE_TABLE	String результат, що вказує ім'я таблиці, яка містить графічні об'єкти. Використовуйте цей код, коли працюєте з таблицею, що є фактично об'єднанням двох інших таблиць, і необхідно знати ім'я основної таблиці, що містить графічні об'єкти.
TAB_INFO_MINX, TAB_INFO_MINY, TAB_INFO_MAXX, TAB_INFO_MAXY	Float результати, які надають мінімум і максимум x-и y-координат із всіх об'єктів у таблиці
TAB_INFO_NAME	String результат, що вказує ім'я таблиці
TAB_INFO_NCOLS	SmallInt, що вказує число стовпців

TAB_INFO_NREFS	SmallInt, що вказує номер іншої основної таблиці, що посилається на цю таблицю. (Нуль повертається для більшості таблиць, або відмінний від нуля у випадках, де таблиця визначена як об'єднання двох інших таблиць, типу StreetInfo таблиці.) Може використовуватись тільки з основними таблицями (TAB_TYPE_BASE).
TAB_INFO_NROWS	Integer, що вказує число рядків.
TAB_INFO_NUM	SmallInt результат, що вказує номер таблиці.
TAB_INFO_READONLY	Logical результат; TRUE, якщо таблиця тільки для читання
TAB_INFO_SEAMLESS	Logical результат; TRUE, якщо властивість зшитої включена для цієї таблиці.
TAB_INFO_TABFILE	String результат, представляє повний шлях до каталогу таблиці. Повертає порожній рядок, якщо таблиця - таблиця запиту.
TAB_INFO_TEMP	Logical результат; TRUE, якщо таблиця тимчасова (наприклад. QUERY1)
TAB_INFO_TYPE	SmallInt результат, що вказує тип таблиці. Повернуте значення буде відповідати одному із цих значень: TAB_TYPE_BASE (якщо нормальна або зшита таблиця) TAB_TYPE_RESULT (якщо результат запиту) TAB_TYPE_IMAGE (якщо таблиця - растрове зображення) TAB_TYPE_VIEW (якщо таблиця - фактичного подання; наприклад, StreetInfo таблиці - фактичного подання) TAB_TYPE_LINKED (якщо ця таблиця зв'язана)

	TAB_INFO_UNDO Logical результат; TRUE, якщо система скасування (undo) використовується із зазначеною таблицею, або FALSE, якщо система скасування була виключена для таблиці інструкцією Set Table
TAB_INFO_USERBROWSE	Logical результат: FALSE, якщо інструкція Set Table установила UserBrowse опцію в Off
TAB_INFO_USERCLOSE	Logical результат: FALSE, якщо інструкція Set Table установила UserClose опцію в Off
TAB_INFO_USERDISPLAYMAP	Logical результат: FALSE, якщо інструкція Set Table установила UserDisplayMap опцію в Off
TAB_INFO_USEREDITABLE	Logical результат: FALSE, якщо інструкція Set Table установила опцію UserEdit в Off
TAB_INFO_USERMAP	Logical результат: FALSE, якщо інструкція Set Table установила опцію UserMap в Off
TAB_INFO_USERREMOVEMAP	Logical результат: FALSE, якщо інструкція Set Table установила опцію UserRemoveMap в Off

Insert statement

Призначення: Додає нові рядки у відкриту таблицю.

Синтаксис: Insert Into table [(*columnlist*)]

{ Values (*exprlist*) | Select *columnlist* From table }

table - ім'я відкритої таблиці;

columnlist - список виражень для стовпців, відділених комами;

exprlist - список одного або більше виражень, відділених комами.

Опис: Інструкція Insert вставляє нові рядки у відкриту таблицю. Є дві головних форми цієї інструкції, які дозволяють або додати один рядок або вставити групу рядків від іншої таблиці (через пропозицію Select) одночасно. У кожному разі, число вставлених значень стовпців повинне відповідати числу стовпців у списку стовпців. Якщо список стовпців не визначений, всі поля приймаються. Зверніть увагу, що необхідно використовувати інструкцію Commit, якщо треба зберегти недавно вставлені записи на диску.

Наступна інструкція створює об'єкт крапка й вставляє об'єкт у новий рядок таблиці Sites. Obj - спеціальне ім'я стовпця, що представляє графічні об'єкти таблиці.

```
Insert Into sites (Obj)  
Values ( CreatePoint(-73.5, 42.8) )
```

Update statement

Призначення: Змінює один або більше рядків у таблиці.

Синтаксис: Update table Set column = expr [, column = expr, ...]

[Where RowID = idnum] table - ім'я відкритої таблиці; column - ім'я стовпця;

expr - вираз, результат якого отримає стовбець;

idnum - номер рядка в таблиці.

Опис: Інструкція Update змінює один або більше стовпців у таблиці. За замовчуванням, інструкція Update торкнеться всіх рядків у зазначеній таблиці. Однак, якщо інструкція включає пропозицію Where Rowid, тільки цей рядок буде модифікований. Пропозиція Set визначає те, які зміни повинні бути зроблені у рядку або рядках. Щоб модифікувати картографічний об'єкт, що пов'язаний з рядком, визначте, стовпець Obj в пропозиції Set.

Контрольні питання

1. Основні типи графічних примітивів для представлення географічних об'єктів.
2. Що таке центроїд об'єкту. Місце центроїда за типами об'єктів.
3. Що таке десяткові градуси.
4. Етапи створення шару географічних даних.
5. Географічні та геометричні характеристики за типами об'єктів.
6. Статистичні характеристики географічних об'єктів.
7. Функції MapBasic для визначення географічних характеристик об'єктів.
8. Функції MapBasic для визначення геометричних характеристик об'єктів.
9. Функції та інструкції MapBasic для визначення статистичних характеристик об'єктів.

Варіанти завдань

1. Створити в MapInfo шар зображення території ділянки карти, згідно

із завданням, приведеним в таблиці 1. Крім того, необхідно додатково створити шари з административно-територіальним діленням областей вказаних в завданні територій. При цьому, атрибутивна частина даних повинна містити такі поля: найменування географічного об'єкту (неповторне); широта центрюду об'єкта; довгота центрюду об'єкта; довжина об'єкту; периметр об'єкту; площа об'єкту.

2. Застосовуючи відповідні вирази до стовбців даних належним чином необхідно заповнити атрибутивні дані.

3. Розробити за допомогою MapBasic програмний додаток який, використовуючи результати виконання першої частини роботи має виконувати наступне: заповнювати відповідні атрибутивні поля розробленого шару об'єктів; в разі вибору того чи іншого об'єкту карти, відповідно до його типу надавати (чи в діалоговому вікні, чи в системному вікні повідомлень) разом географічні, геометричні та статистичні характеристики цього об'єкту; автоматично створювати та записувати в текстовий файл таблиці 1, 2

Растрове зображення (рис.9) ділянки карти, з вказівкою населених пунктів, представлено у файлах *Europe(z8).tab* и *Europe(z8).jpg*.

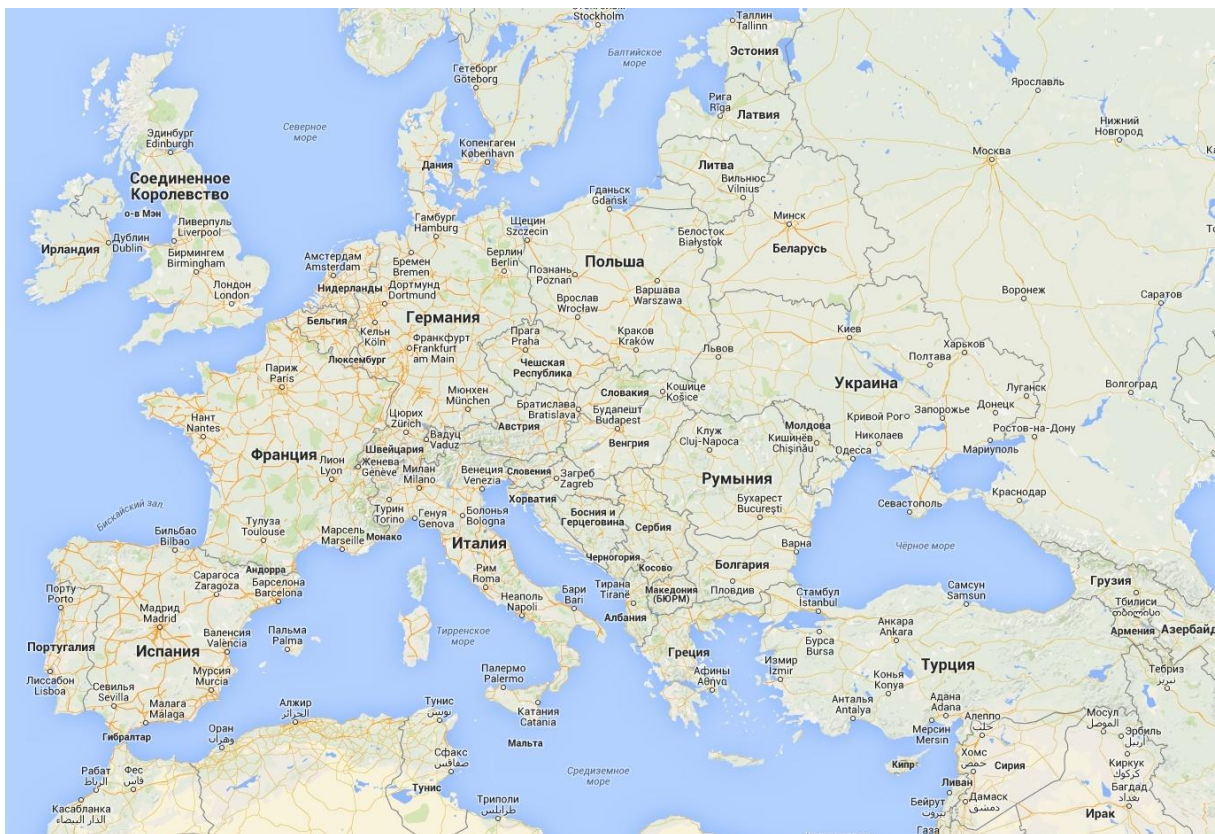


Рис. 9 Карта територій

Таблиця 1. Варіанти завдань

№ варіанта	Наименование территории (государства)	Примечание
1	Румунія	Бухарест
2	Угорщина	Будапешт
3	Чехія	Прага
4	Італія	Рим
5	Франція	Париж
6	Німеччина	Берлін
7	Іспанія	Мадрид
8	Греція	Афіни
9	Польща	Варшава
10	Австрія	Відень
11	Болгарія	Софія
12	Туреччина	Стамбул

Завдання № 3

Тема: Тематичні карти

Мета роботи

Одною з основних переваг геоінформаційних технологій є можливість подання або відтворення будь-якого просторово розподіленого показника за допомогою карт. Можливість зберігання атрибутивної інформації в геоінформаційних базах надає можливість проведення різноманітних видів просторового аналізу як на основі цих атрибутивних даних, так і на базі обчислених за цими даними показників. Подальше представлення такої інформації в картографічному вигляді є найбільш важливою складовою геоінформаційних систем. Основними інструментами систем, яки надають такі можливості є інструменти побудови тематичних карт.

На відміну від цифрової картографії, націленої на створення усе більш точних цифрових (електронних) аналогів загальногеографічних і топографічних карт, ГІС-картографування носить яскраво виражений інформаційноаналітичний і евристичний характер. Тому воно в набагато меншому ступені орієнтовано на точне відтворення картометричних характеристик території (карто- і топооснов), і акцентує увагу користувача на закономірності просторового розподілу атрибутивних (тематичних)

ознак. Найчастіше така орієнтація приводить до створення комп'ютерних псевдо- і квазикарт - електронних аналогів різних картосхем, картограм, картодіаграм. У процесі тематичного ГІС-картографування фактично використовується не метрична, а топологічна модель території, що обумовлено віртуальним (тимчасовим, ситуаційним) характером використання електронних тематичних карт у процесі прийняття рішень. У цьому випадку картометричні характеристики, як правило, не попадають у поле зору користувача тематичної карти, якого більше цікавлять топологічні аспекти розподілу тематичних ознак (наприклад, сусідство, форма, характер розподілу таксонів по градаціях тематичної ознаки).

Тематичним картографуванням (thematic mapping) називають комплекс заходів і процесів по створенню *тематичних карт і атласів*.

Як розділи тематичного картографування виділяють картографування природи (геологічне, кліматичне, ґрунтове, геоботанічне й ін.) суспільства (населення, господарства, історичне і т.п.) і їхньої взаємодії (інженерно-геологічне, екологічне, природоохоронне й ін.).

За практичною спеціалізацією тематичне картографування може бути інвентаризаційним, оцінним, прогнозним, рекомендаційним, а за рівнем узагальнення – аналітичним, комплексним і синтетичним. Найбільша синтетичність і різнобічність притаманна *системному картографуванню*

Аналітичне картографування (analytical mapping) – створення вузькоспеціалізованих карт, для окремо досліджуваного явища чи показника.

Комплексне картографування (complex mapping) – багатобічне, цілісне картографічне відображення дійсності. Виконується на системній основі, його результатом є серії тематичних карт чи комплексні атласи (complex atlases), що характеризують природу, населення, господарство і їхню взаємодію. Карти, що входять у серію чи атлас, відрізняються погодженістю і взаємною доповнюваністю, що забезпечує зручність комплексного вивчення території.

Синтетичне картографування (synthetic mapping) – один з напрямків *тематичного картографування*, у якому розробляються теорія і методи створення *синтетичних карт* на основі інтеграції безлічі приватних показників і (чи) серій *аналітичних і комплексних карт*. Широко спирається на методи факторного аналізу, дискримінантного аналізу, виділення головних компонентів, кластеризацію й ін., методи *математико-картографічного моделювання*, що дозволяють одержувати інтегральні характеристики картографуємих об'єктів.

Системне картографування (system mapping) – один з науково-технічних напрямків картографії, що включає системне створення і використання картографічних добутоків як моделей геосистем. Припускає моделювання геосистем, їх компонентів, взаємозв'язків, ієрархії, динаміки і функціонування в системі карт. Принципи системного картографування

знаходять найбільш повне вираження в комплексних науково-довідкових атласах і серіях тематичних карт.

Тематична карта (thematic map) – син. галузева карта – карта, що відбиває який-небудь один сюжет (тему, об'єкт, явище, галузь) чи сполучення сюжетів. Розрізняють тематичні карти природних, суспільних явищ і їхньої взаємодії (напр., карти геологічні, етнографічні, соціально-економічні, екологічні і т.п.). За ступенем узагальнення зображуваних явищ виділяють *аналітичні, комплексні і синтетичні* карти.

Аналітична карта (analytical map) – карта, що показує неузагальнені чи малоузагальнені показники якого-небудь явища (напр., карта температури повітря) чи тільки окремі сторони об'єкта (напр., карта експозиції схилів рельєфу).

Комплексна карта (complex map, aggregate map) – карта, що показує спільно кілька різних взаємозалежних явищ (чи кілька властивостей одного явища), але кожне у своїй системі показників.

Синтетична карта (synthetic map) – карта, що дає інтегральне зображення об'єкта чи явища в єдиних синтетичних показниках. Найчастіше синтетичні карти відбивають типологічне районування території за комплексом показників (напр., ландшафтне, кліматичне районування, розподіл території за умовами життя населення і т.п.).

Картограма (choropleth map, cartogram, chorogram, chorisogram)

- 1. карта, що показує розподіл відносних показників (щільність, інтенсивність якогонебудь явища, питомі величини і т.п.) по визначених територіальних одиницях;

- 2. один зі способів картографічного зображення, застосовуваний для показу відносних статистичних даних шляхом заповнення контурів територіального розподілу колірними заливаннями (solid) різного тону, штрихуваннями (crosshatch line pattern) різної щільності відповідно до прийнятих інтервальних шкал. Засоби автоматизації дозволяють будувати картограми в так званих безупинних чи безінтервальних шкалах (choropleth maps without class intervals, continuous tone cartogram), коли щільність ставиться в точну відповідність величині картографуємого показника.

Картодіаграма (diagram map, diagrammatic map)

- 1. карта, що відбиває розподіл якого-небудь явища за допомогою діаграм: лінійних, чи стовпчастих картограм (bar chart), площадних картограм (area chart) чи об'ємних картограм (3D bar chart), локалізованих по одиницях територіального розподілу;

- 2. один зі способів картографічного зображення, що використовується для показу абсолютних статистичних даних.

Теорія і методи аналізу геообразень складають предмет нового наукового напрямку - *геоіконіки*, що зв'язує в один науково-технологічний комплекс картографію, дистанційне зондування і машинну графіку.

Важливо також зауважити, що розбудова тематичної карти в геоінформаційній системі займає набагато менше часу, ніж традиційно, при наявності геоінформаційної бази, що дозволяє проводити просторовий аналіз досить швидко, в тому числі і в реальному часі. Можливості ГІС в плані розбудови тематичних карт є одним з найважливіших показників при визначенні переваг тієї або іншої геоінформаційної системи.

Метою даної лабораторної роботи є ознайомлення студентів з можливостями та особливостями роботи з розбудови різноманітних типів тематичних карт при вирішенні питань, що торкаються аналізу просторових даних.

Завдання до самостійної роботи

Завдання – надати необхідні знання та виробити навички у студентів щодо роботи з тематичними картами.

Після виконання завдання студенти повинні:

- *знати* типи тематичних карт, шляхи та методи їх побудови;
- *вміти* різними способами отримувати необхідні тематичні карти, вибирати та обґрунтовувати оптимальні типи тематичних карт для вирішення того чи іншого питання.

Робота спирається на знання й уміння, отримані при вивченні "Основ СУБД", виконанні попередніх лабораторних робіт.

За завданням викладача студенту необхідно розробити шари тематичних карт, обґрунтувавши вибір того чи іншого типу тематичної карти. За отриманими картами провести письмовий аналіз згідно завданню.

Методичні вказівки

Розробка тематичних карт засобами системи. Відповідно до завдання необхідно засобами системи розробити шари тематичних карт та надати їх письмовий аналіз. Відповідна географічна база надається викладачем разом з завданням. Розробка здійснюється в ручному режимі з використанням відповідних інструментів системи. Якщо в системі відкритий хоча б один географічний шар векторних об'єктів, система, через меню "Карта – Створити тематичну карту", надає можливість розробити тематичну карту (рис.1).

Процес створення тематичної карти в системі складається з трьох шагів, чи етапів. На першому кроці вибирається тип тематичної карти. Загалом система пропонує сім різновидів тематичних карт. Слід зауважити, що так чи інакше ці типи вкладаються в два типи тематичних карт, згаданих в п.1. Кожен різновид має певний перелік збережених зразків(стилів) тематичних карт. Система дозволяє користувачу поповнювати цей перелік своїми зразками, про що буде згадано пізніше. Розглянемо різновиди тематичних карт з точки зору доцільності їх

використання при аналізі тих чи інших питань.

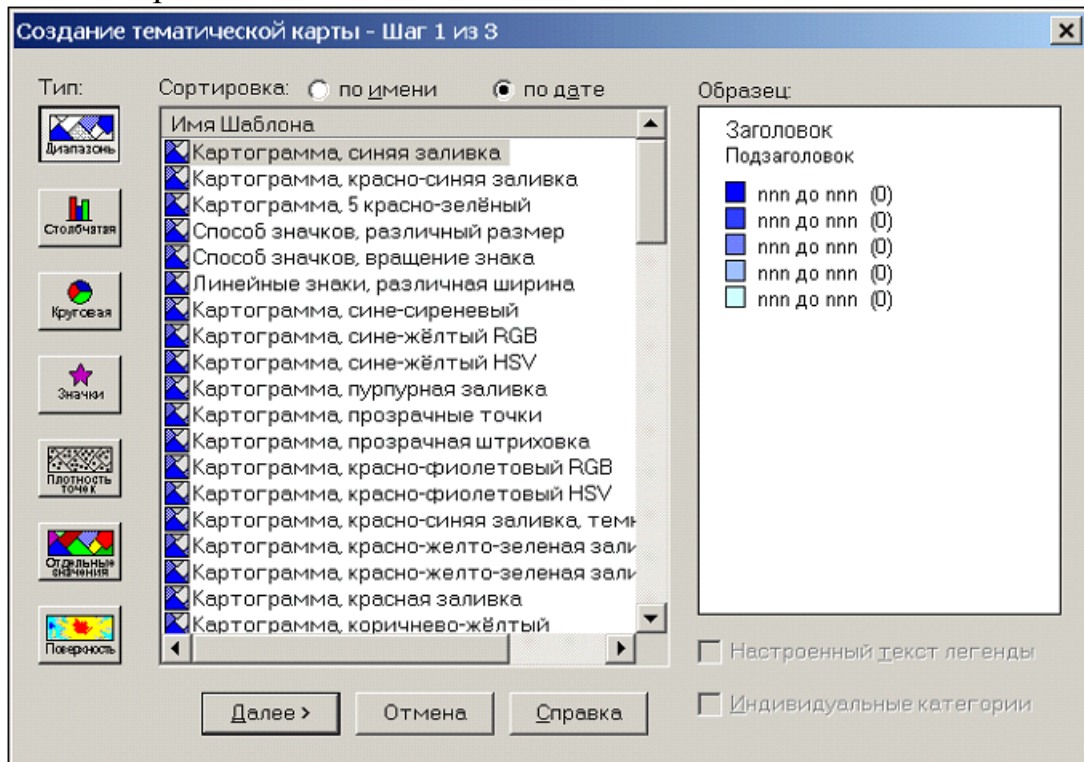


Рис.1. Діалогове вікно першого шагу створення тематичної карти

- 1) “Ranges”. Картограми за інтервалами значень. Цей різновид тематичних карт доцільно використовувати при оцінці параметрів чи показників що мають велику розбіжність та за якими необхідно мати уявлення з просто- рового розподілу по територіях. Частіше використовуються в аналітичних картах для інтегральних показників, які класифікуються за інтервалами значень та характеризують собою певні умови на території (наприклад, індекс забруднення атмосфери, бонітет, середня ціна нерухомості, тощо).
- 2) “Bar Charts”. Картодіаграма з використанням стовпчикових діаграм. Доцільно використовувати при аналізі за декількома параметрами чи показниками одночасно. Хоча ці параметри і пов’язані з тією чи іншою територією, та вони не потребують при аналізі співставлення охоплення територій, наприклад, населення держав (порівняйте – населеність держав).
- 3) “Pie Charts”. Картодіаграма з використанням кругових діаграм. За доцільністю використання відповідають стовпчиковим діаграмам. Найбільш часто використовуються для побудови тематичних карт з урахуванням різних показників за галузевим принципом. Крім того, на відміну від стовпчикових діаграм, кругові надають можливість оцінки суми параметрів що розглядаються. Тому, крім галузевого, доцільно використовувати цей різновид тематичних карти при

оцінці ситуації за сумою показників, якщо така сума має сенс та доцільна, тобто для комплексних та синтетичних карт. Наприклад, якість життя, загальна захворюваність населення та інше.

- 4) “Graduated”. Це різновид картодіаграми з використанням символів. На відміну від стовпчикових та кругових діаграм в цих картах використовуються символи, які характеризують змістовно атрибут, за яким будується тематична карта, та розміром відповідають значенню цього атрибута. Слід зауважити, що цей різновид тематичних карт вимагає наявності бібліотеки символів, та потребує досить великої їх різноманітності за розміром. Тому цій різновид тематичних карт не таких розповсюджений як три попередні. Доцільно використовувати як в аналітичних так і в комплексних картах (при одночасному відображенні декількох атрибутів).
- 5) “Dot Density”. Картограма за щільністю точок. Цей різновид тематичних карт можуть використовуватись у випадках, що й картограми за інтервалами значень, але найбільш доцільно їх застосовувати для характеристик щільності (щільність населення, щільність аптечної мережі, таке інше).
- 6) “Individual”. Картограма за окремими значеннями. Цей тип тематичної карти дозволяє будувати карту не за інтервалом значень якогось, в тому числі й розрахункового параметру, як всі попередні різновиди, а по кожному значенню цього параметра. Доцільно використовувати в аналітичних картах.
- 7) “Grid”. Растрова карта інтерпольованих значень. Цей вид карт притаманний лише векторним ГІС-пакетам. Одним з недоліків векторного представлення даних в ГІС є вимушена дискретизація безперервних величин або параметрів. До того ж, не завжди є можливість отримувати поля тих чи інших величин на тій чи інших території. Тому, якщо дозволяє характер величин, їх безперервні поля можна отримати інтерполюючи отримані в окремих точках поля значення. Для цього й призначений різновид “Grid”. Тут той чи інший показник, за яким будується карта, не пов’язаний з географічним об’єктом. Цей різновид надає лише просторовий розподіл (поле) того чи іншого показника. При цьому можуть використовуватись різні методи інтерполяції, наприклад, пакет MapInfo використовує зважену середню відстань між точками даних для обчислення значення ячейки растру, тобто IDW (Inverse Distance Weighting) інтерполяцію та TIN (Triangulated Irregular Network) інтерполяцію; пакет Surfer, хоч це і не ГІС-пакет, використовує для цього більше п’яти методів.

Вибір того чи іншого типу тематичної карти залежить, крім іншого, і від типу географічних об’єктів, для яких розбудовується карта.

Картограми за інтервалами значень, можна застосовувати до всіх типів об'єктів, застосовуючи або колір, або розмір для позначення.

Оновлення колонки тематичних карт. Як вже говорилося в цій главі, для побудови тематичних Карт можна використати дані з інших таблиць. Для цього слід вибрати Об'єднання з віконця списку в другому діалозі створення тематичних Карт. MapInfo Professional покаже діалог

Операція оновлення колонки створює тимчасову колонку в початковій таблиці і автоматично заповнює її даними. Ці дані можуть бути безпосередньо перенесені з іншої таблиці, або отримані в результаті Обчислення чи узагальнення даних.

Щоб створити тимчасову колонку командою Оновити колонку:

1. Відкрийте обидві таблиці - базову і ту звідки вимагається узяти дані
2. Виконайте команду Карта > Створити тематичну Карту. З'явиться діалог Створення тематичної Карту - Крок 1 з 3.
3. Натисніть Діапазони і виберіть назву шаблону зі списку. Це стиль створюваної тематичної Карту. Натисніть на кнопку Далі для переходу в діалог Створення тематичної Карту - Крок 2 з 3.
4. Виберіть таблицю районів, У віконці списку Поле виберіть Об'єднання. З'явиться діалог Оновити тематичну колонку.

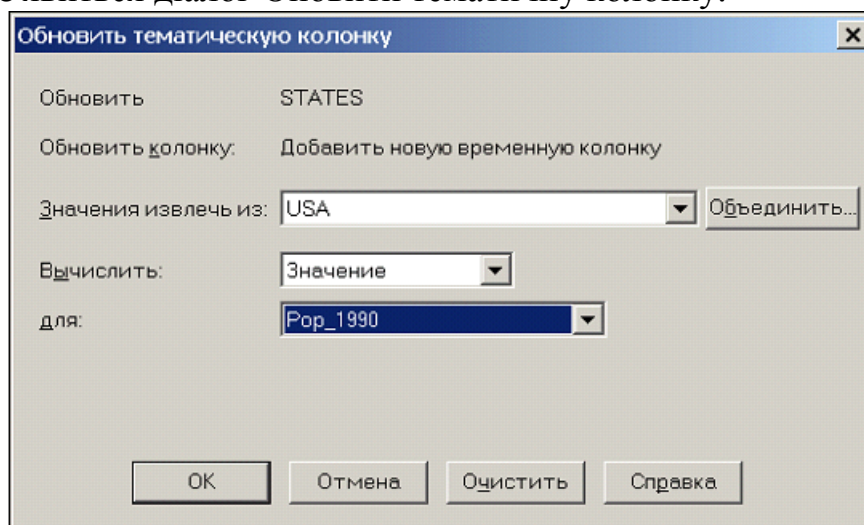


Рис.2. Діалогове вікно «Оновити тематичну колонку»

5. Натисніть кнопку Об'єднати в діалозі Оновити тематичну колонку.

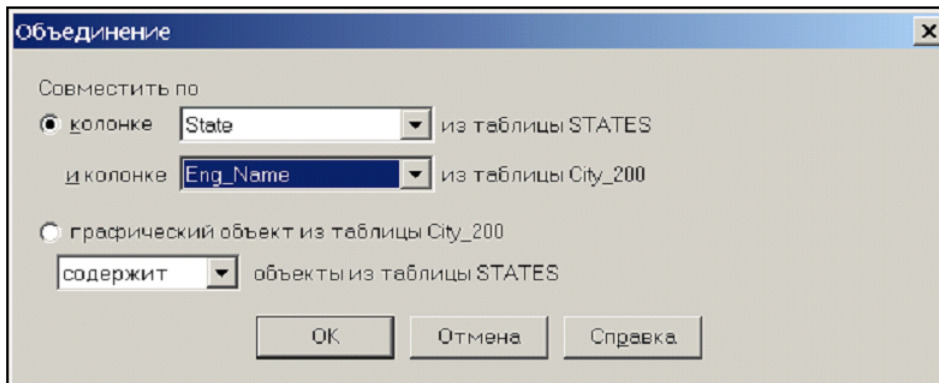


Рис.3. Діалогове вікно «Об'єдння»

Коли вибираєте Об'єднати з діалогу Оновити тематичну колонку, то побачите, що MapInfo Professional вже вибрала в діалозі Об'єднання однакові колонки для обох таблиць. Якщо проводиться об'єднання там, де не проходить автоматичне обчислення, але необхідно вказати колонки по яких буде робиться зіставлення, або звернутися до географічного об'єднання.

6. Натисніть кнопку ОК. MapInfo Professional порахує суми і повернеться в діалог Створення тематичної Карти - Крок 2 з 3. У віконці Поле Ви побачите створену тимчасову колонку "СуммаНаличное".

Усі дані підготовлені, за винятком процентного співвідношення. Оскільки немає поля в таблиці, що містить цю величину, потрібно створити вираження для обчислення цього процентного співвідношення.

7. У діалозі Створення тематичної Карти - Крок 2 з 3 в списку Поле виберіть Вираження. З'явиться діалог Вираження.

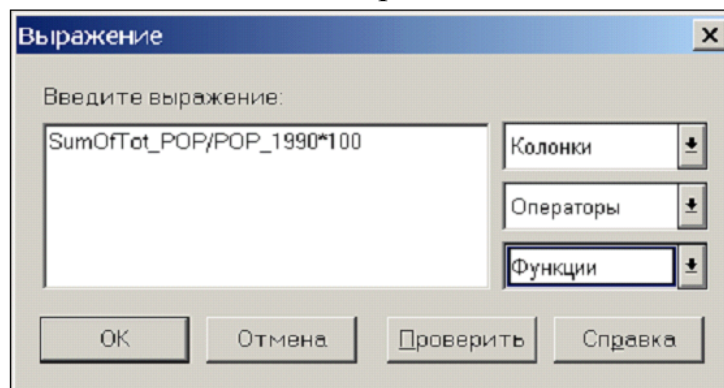


Рис.4. Діалогове вікно конструктора виразів

8. ідсоток міського населення обчислюється вираженням "СуммаНаличное/Населення * 100".

У вікні Введіть вираження, наберіть:

СуммаНаличное/Население * 100

9. Натисніть кнопку **Перевірити**, щоб перевірити правильність синтаксису, а потім - кнопку **ОК**. Ви знову повернетесь в діалог Створення тематичної Карти - Крок 2 з 3, і в кінці Поле міститиме тільки що створене вираження. Натисніть **Далі**, щоб перейти до діалогу Створення тематичної Карти - Крок 3 з 3.

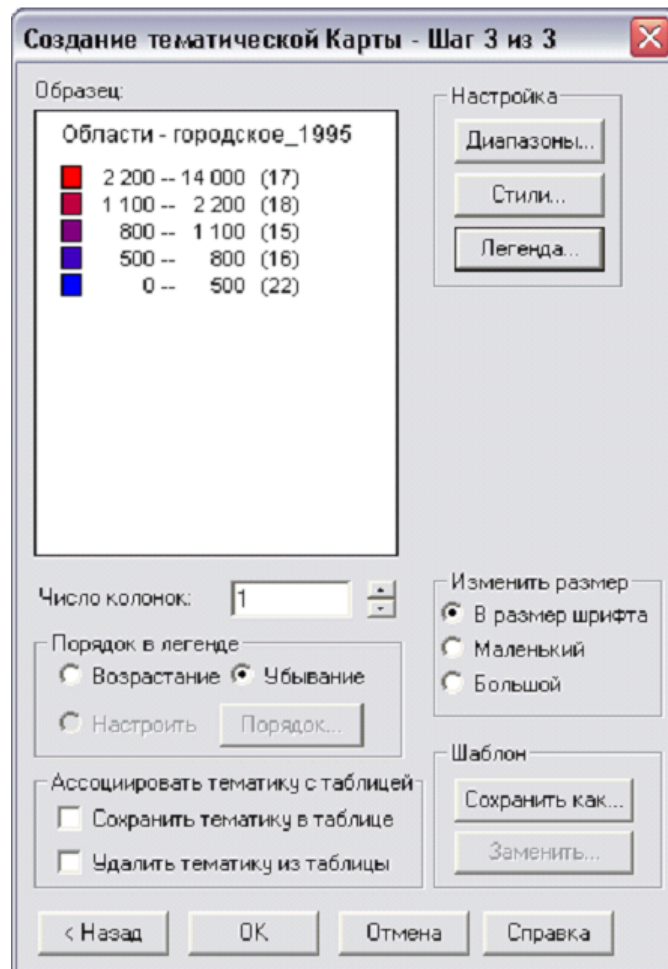


Рис.5. Діалогове вікно «Створення тематичної Карти»

10. Виберіть режим Рівна кількість записів або Рівний розкид значень для діапазонів залежно від того, який із способів краще відобразить розподіл даних. При необхідності використовуйте округлення значень. Налаштуйте стилі по своєму бажанню (використайте кнопку **Стиль**). Налаштуйте легенду, натиснувши кнопку **Легенда**.

11. Натисніть **ОК** для показу Карты у вікні Карты.

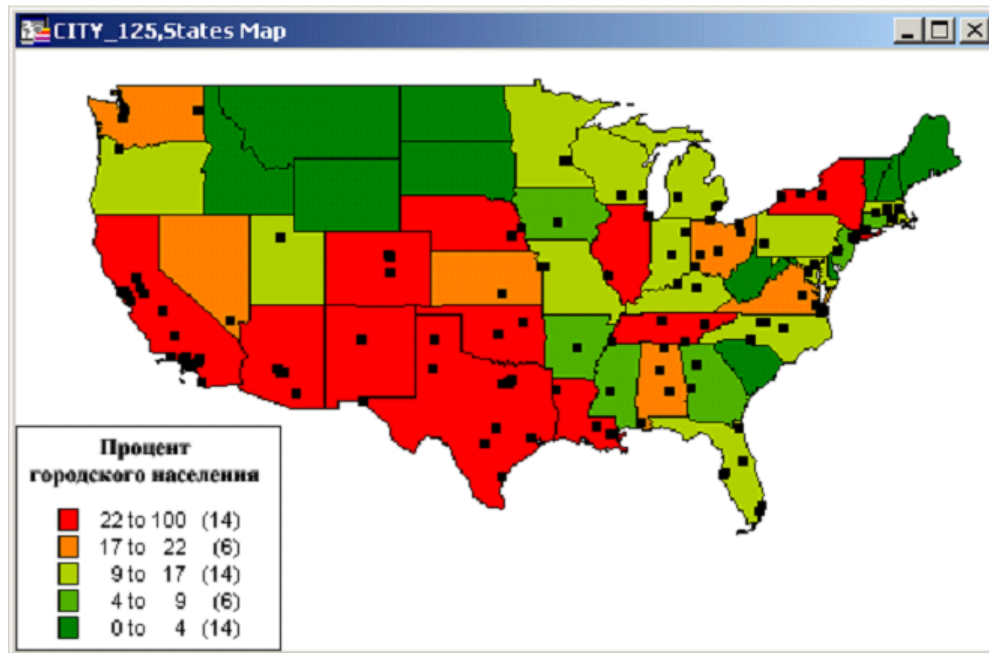


Рис.6. Вікно тематичної карти

Контрольні питання

1. Що таке тематичні карти.
2. Методи тематичної картографії.
3. Типи тематичних карт.
4. Основні етапи створення тематичних карт.
5. Що таке тематична легенда.

Варіанти завдань

1. Розробити шари тематичних карт, обґрунтувавши вибір того чи іншого типу тематичної карти. Розробку шарів провести відобразивши населення для країн світу, що вказані в таблиці відповідно до варіанта. За отриманими картами провести письмовий аналіз згідно завданню.

Таблиця 1. Варіанти завдань

№ варіанта	Наименование территории (государства)
1	Румунія, Болгарія, Італія
2	Угорщина, Чехія, Словаччина
3	Хорватія, Чехія, Албанія
4	Італія, Німеччина, Португалія
5	Франція, Іспанія, Німеччина

6	Данія, Франція, Румунія
7	Австрія, Польща, Португалія
8	Австрія, Польща, Португалія
9	Польща, Австрія, Німеччина
10	Туреччина, Сирія, Іран
11	Греція, Польща, Україна
12	Україна, Молдавія, Грузія

Примичание: В якості початкових даних використати файл world.tab та і його компоненти.

3. ОРГАНІЗАЦІЯ КОНТРОЛЮ ЗНАНЬ ТА ВМІНЬ СТУДЕНТІВ

3.1. Система контролю знань та вмінь студентів

Контроль знань та вмінь студентів, що навчаються за заочною формою, здійснюється за допомогою системи контролюючих заходів. Вони складаються з заходів поточного та підсумкового контролю. Поточний контроль здійснюється на протязі всього навчального року (семестру) та включає заходи контролю самостійної роботи студента під час вивчення навчальної дисципліни поза межами університету та роботи студента на лабораторних заняттях у період заліково-екзаменаційної сесії.

Підсумковий контроль здійснюється під час заліково-екзаменаційної сесії та має на меті встановлення рівня знань та вмінь, які опанував студент після вивчення навчальної дисципліни. Форма підсумкового контролю - іспит - встановлюється навчальним планом дисципліни.

При заочній формі вивченні дисципліни „Геоінформатика і ГІС» використовується накопичувальна система оцінювання. Форма підсумкового контролю – залік. Головними рисами накопичувальної системи є підсумкова оцінка знань студента складається як арифметична сума оцінки, яку накопив студент, виконуючи заходи поточного контролю, та оцінки, яку отримав студент на підсумковому контролі (заліку);

3.2 Форми контролю знань та вмінь студентів

1) Оцінка самостійної роботи студента до екзаменаційно-залікової сесії здійснюється у формі оцінки виконання самостійних робіт.

Максимальний бал за контрольну роботу становить 15 балів, з них на теоретичну частину припадає 5 бали і на практичну частину – 10 балів за кожне завдання.

Контрольна робота вважається зарахованою, якщо студент одержав сумарну оцінку не менше 9 балів (3 балів з теоретичної частини і 6 балів з практичної частини), тобто не менше 60% від максимальної суми в 15 балів. Не зарахована контрольна робота свідчить про те, що студент одержав сумарну оцінку меншу за 9 балів, в цьому випадку вона повертається на доробку.

Оцінка виконання курсової роботи включається окремою складовою у накопичувальну оцінку виконання студентом практичної частини дисципліни. Зараховані контрольна є допуском до здачі заліку.

2) Оцінки роботи студента при проведенні лабораторних занять по дисципліні під час екзаменаційно-залікової сесії.

Загальна максимальна оцінка роботи студента під час заліково-екзаменаційної сесії складає 40 балів і включає: за виконання контрольної роботи – 15 балів, захист виконаних лабораторних робіт – 15 балів, залікова контрольна – 10 балів.

Суми балів, які отримав студент за всіма змістовними модулями (заліковими одиницями) навчальної дисципліни, формують інтегральну оцінку поточного контролю студента з навчальної дисципліни. Вона є підставою для допуску студента до семестрового іспиту. Студент вважається допущеним до підсумкового семестрового контролю з дисципліни, якщо він виконав всі види робіт, передбачені робочою навчальною програмою дисципліни і набрав за модульною системою суму балів не менше 50% від максимально можливої за практичну частину дисципліни для заліку, передбаченого робочим навчальним планом та робочою програмою дисципліни.

- Кількісна оцінка заходу підсумкового контролю (ОПК)

Цей захід передбачає оцінювання результатів іспиту, який виконується в період заліково-екзаменаційної сесії.

Студент, який не має на початок заліково-екзаменаційної сесії заборгованості по дисципліні, допускається до залікової контрольної роботи за затвердженим розкладом та процедурою.

Бали успішності (у відсотках), які студент отримав за підсумками іспитів переносяться до графі 4 заліково-екзаменаційній відомості.

Максимальна кількість балів за кожне питання та по заліковому тесту в цілому, доводиться до відома студентів на початку семестру. Кожна правильна відповідь відповідає 0,4 балу.

Таблиця для визначення кількісних показників успішності за підсумками заліку.

Кількість балів за залікт	За системою університету (у відсотках)	Визначення
36 – 40	90 – 100	відмінне виконання лише з незначною кількістю помилок
34 – 35	85 – 89	вище середнього рівня з кількома помилками
30 – 33	75 – 84	в загальному правильна робота з певною кількістю грубих помилок
27 – 29	68 – 74	непогано, але зі значною кількістю помилок
24 – 26	60 – 67	виконання задовольняє мінімальні критерії
14 – 23	35 – 59	з можливістю перескласти
1 – 13	1 – 34	з обов'язковим повторним курсом навчання

- Система накопичувальної підсумкової оцінки засвоєння студентами початкової дисципліни (ПО)

Накопичена підсумкова оцінка засвоєння студентами заочної форми навчальної дисципліни розраховується, як:

$$ПО = 0,5ОПК + 0,25(ОЗЕ + ОМ).$$

Загальна підсумкова оцінка знань студента з іспиту складається як арифметична сума, яку накопив студент у міжсесійний період, під час заліково-екзаменаційної сесії та складання іспиту. Таким чином студент може одержати максимально 100 балів.

Якісна оцінка є такою:

- 90 і більше – відмінно;
- 76-89 балів – добре;
- 60-74 балів – задовільно;
- менше 60 балів – незадовільно.