Open Geospatial Consortium Inc.
Дата: 2006-10-05
Учетный номер документа: OGC 06-103r3
Версия: 1.2.0
Категория: OpenGIS® Implementation Specification
Редактор: John R. Herring
Перевод: Бондарец А.В., Григорива О.М, Рыков Д.А., Дубинин М.Ю.
Перевод выполнен на базе GIS-Lab.info

OpenGIS® Implementation Specification for Geographic information - Simple feature access - Part 1: Common Architecture

Спецификация реализации пространственной информации OpenGIS® – Доступ к простой геометрии - Часть 1: Общая архитектура

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.
Дополнительная информация о правах на использование: http://www.opengeospatial.org/legal/

Тип документа: OpenGIS® Implementation Specification
Ссылка на оригинал: http://www.opengeospatial.org/standards/sfa
Подтип документа: (нет)
Уровень: Кандидат
Язык документа: Русский

Содержание

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Рисунки

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Предисловие от переводчиков

This translation of the OGC document OpenGIS® Implementation Specification for Geographic information - Simple feature access - Part 1: Common Architecture is informative. The English version of document OGC document OpenGIS® Implementation Specification for Geographic information - Simple feature access - Part 1: Common Architecture found at the OGC web site is the normative reference.

Этот перевод документа OGC Спецификация реализации пространственной информации OpenGIS® – Доступ к простой геометрии - Часть 1: Общая архитектура носит информационный характер. Английская версия документа OpenGIS® Implementation Specification for Geographic information - Simple feature access - Part 1: Common Architecture расположенная на сайте OGC носит нормативный характер.

Данный перевод предоставляется «как есть», может, и, скорее всего, имеет ошибки привнесенные в процессе перевода.

Переводчики и редакторы: Бондарец А.В., Григорива О.М, Рыков Д.А., Дубинин М.Ю. Перевод выполнен на базе GIS-Lab.info. Версия документа 1.0.

Постоянная ссылка на этот документ: http://gis-lab.info/docs/ogc-sfa1-v1.pdf
Ссылка на оригинал: http://portal.opengeospatial.org/files/?artifact_id=18241

Примечания переводчиков даны сносками.

Предисловие

Обращаем внимание на возможность того, что некоторые из элементов этого документа могут быть субъектом патентного права. В задачи OGC не входит явное обозначение любых таких случаев.

Эта спецификация состоит из двух частей, под общим названием *Географическая информация—Доступ к простой геометрии*:
— *Часть 1: Общая архитектура*
— *Часть 2: Опция SQL*

Эта версия заменяет все предыдущие версии Спецификации OpenGIS® Simple Features Implementation Specification, включая частично OGC 99-049 "OpenGIS Simple Features Specification for SQL Rev 1.1", OGC 99-050 "OpenGIS Simple Features Specification For OLE/COM Rev 1.1", OGC 99-054 “OpenGIS Simple Features Specification For CORBA Revision 1.1.”, и OGC 05-126 “OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture”.

Версия 1.1 этой спецификации – конспект настоящей версии, включающий частичное описание технологии из этого документа и не включающая исправления и уточнения.

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Введение

Эта часть OpenGIS® Simple Features Access (SFA), также известного как ISO 19125, описывает общую архитектуру для простых геометрических объектов. Модель простых геометрических объектов является нейтральной по отношению к Распределенной Вычислительной Платформе (Distributed Computing Platform) и использует нотацию UML. Базовый класс Геометрия (Geometry) имеет подклассы для Точки, Кривой, Поверхности и Коллекции (Point, Curve, Surface, GeometryCollection). Каждый геометрический объект ассоциирован с пространственной системой координат, описывающей координатное пространство в котором определен этот объект.

Расширенная модель Geometry описывает 0, 1 и 2-мерные классы Мультиточка (MultiPoint), Мультиломаная (MultiLineString) и Мультиполигон (MultiPolygon), используемые для моделирования геометрических объектов, представляющих из себя несколько Точек, Ломаных или Полигонов. Мультикривая (MultiCurve) и Мультиповерхность (MultiSurface) [1] введены как абстрактные суперклассы организующие доступ к коллекциям Curve и Surface.

Свойства, методы и утверждения для каждого класса Geometry описаны в 6.1.1 (Рисунок 1). Методы описываются по отношению к целевом объекту (объкт которому посылается сообщение).

Функции SFA COM могут использовать нотацию отличную от SFA SQL. Нотация COM более знакома программистам, использующим COM. Однако, в этой части OGC 05-126 используется нотация UML. В разных частях этой Спецификации методы могут отличаться. В этом случает приводится описание различий. Эта часть OGC Simple Feature Access вводит профиль пространственной схемы, описанный в ISO 19107:2003, *Geographic information - Spatial schema*. Приложение A детально описывает оответствие схемы использованой в SFA схеме, описанной в ISO 19107:2003.

[1] Далее используются английские названия классов и подклассов (прим. перев.)

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Географическая информация — Доступ к простым объектам Часть 1: Общая архитектура

1 Охват

Эта спецификация устанавливает общую архитектуру и определяет термины, используемые в этой архитектуре. Эта спецификация не пытается стандартизировать и не зависит ни в коей мере от механизма, с помощью которого добавляются и поддерживаются Типы, включая:

а) синтаксис и функциональность для определения типов;
б) синтаксис и функциональность для определения функций;
в) физическое хранение экземпляров типа в базе данных;
г) специфическую терминологию, используемую для обращения к пользовательским типам (User Defined Types), например UDT.

Эта спецификация стандартизует имена и геометрические определения для Типов Geometry.

Эта спецификация не накладывает ограничений на то, как определять Типы Geometry во внутренних схемах, а так же на то когда, как и кто их определяет.

2 Соответствие

Чтобы соответствовать этой спецификации, продукт должен проходить один или больше наборов тестов предлагаемых в ISO 19125.

3 Нормативные ссылки

Примение этой спецификации невозможно без учета нижеперечисленных документов. Для датированных документов, применима только цитированная редакция. Для документов без даты, применима последняя редакция (включая все поправки).

[1] ***ISO/IEC CD 13249-3:2006(E) — Text for FDIS Ballot Information technology — Database languages – SQL Multimedia and Application Packages — Part 3: Spatial, May 15, 2006.***
[2] ***ISO 19107, Geographic information — Spatial schema***
[3] ***ISO 19111, Geographic information — Spatial referencing by coordinates***
[4] ***ISO 19133, Geographic information — Location based services — Tracking and navigation***

4 Термины и определения

В рамках этого документа, применяются все определения из Части 1 этой спецификации и следующие термины.

4.1
граница (boundary)
набор, представляющий ограничения объекта

ПРИМЕЧАНИЕ: Граница наиболее часто используется в контексте геометрии, где набор – коллекция точек или объектов, представляющих эти точки. В других областях термин может использоваться метафорически для понимания перехода между объектом и всем остальным.
[ISO 19107]

4.2
буфер (buffer)
геометрический объект (geometric object) (4.14) содержащий все **direct positions** (4.8) чьё расстояние до указанного геометрического объекта меньше или равно заданному расстоянию
[ISO 19107]

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

4.3
координата (coordinate)
одно из последовательности n-чисел определяющее положение точки (**point)** (4.17) в n-мерном пространстве

ПРИМЕЧАНИЕ: В системе координат, числа должны иметь единицы измерений.
[адаптировано из ISO 19111]

4.4
измерение координат (coordinate dimension)
число измерений или осей необходимых для описания положение в системе координат (**coordinate system)** (4.6)
[ISO 19107]

4.5
референцная система координат (coordinate reference system)
система координат (coordinate system) (4.6) соотносящаяся с реальным миром с помощью датума

[адаптировано из ISO 19111]

4.6
система координат (coordinate system)
набор математических правил определяющих как точке (**point)** (4.17) назначаются координаты (**coordinates)** (4.3) [ISO 19111]

4.7
кривая (curve)
топологический 1-мерный геометрический примитив (**geometric primitive)** (4.15), представляющий непрерывное изображение линии

ПРИМЕЧАНИЕ: Граница кривой – набор точек на концах кривой. Если кривая замкнута, т.е. два ее конца идентичны, и кривая топологически замкнута, то считается, что у нее нет границы. Первая точка называется начальной, последняя – конечной точкой. Связность кривой гарантируется условием «непрерывности». Топологическая теорема гласит, что непрерывное изображение связанного набора так же связанно.

Термин “1-мерная” относится к топологическому измерению примитива. В этом случае это означает, что каждая точка не на границе является членом топологически открытого набора внутри кривой, изоморфичной открытому интервалу (0, 1). Для этой спецификации, измерений координат может быть 2 (для x и y), 3 (с добавленным z или m), или 4 (добавлены оба z и m). Измерения x, y и z пространственные, а m - измерение.
[ISO 19107]

4.8
Прямое местоположение (direct position)
Положение описанное единым набором координат (**coordinates)** (4.3) в системе координат (**coordinate reference system)** (4.5)
[ISO 19107]

4.9
конечная точка (end point)
последняя **точка** (4.17) **кривой** (4.7)

ПРИМЕЧАНИЕ: Конечная и начальная точка относятся к ориентации кривой. Любое представление кривой может быть “перевернуто”, заменив конечную и начальную точки, без изменения изображения кривой как набора точек (прямых местоположений).
[ISO 19107]

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

4.10
внешнее (exterior)
разница между пространством (universe) и закрытым объектом (closure)
ПРИМЕЧАНИЕ: Концепция внешнего применима и к топологическим и к геометрическим комплексам.
[ISO 19107]

4.11
объект (feature)
абстракция явления реального мира

ПРИМЕЧАНИЕ: Объект может проявляться как тип или как экземпляр. Тип объекта или экземпляр объекта используются каждый в своём случае.
[адаптировано из ISO 19101]

4.12
атрибут объекта (feature attribute)
характеристика **объекта** (4.11)

ПРИМЕЧАНИЕ: Атрибут объекта имеет имя, тип данных и домен значений. Атрибут для экземпляра объекта берет некое значение из домена значений. На типы атрибутов не накладывается никаких ограничений. Геометрии, ассоциированные с объектами – один из типов возможных атрибутов объекта.
[адаптировано из ISO 19101]

4.13
геометрический комплекс (geometric complex)
набор раздельных **геометрических примитивов** (4.15) в котором **граница** (4.1) каждого примитива может быть представлена как общее других геометрических притивов меньших измерений в рамках того же набора

ПРИМЕЧАНИЕ: Геометрические примитивы в набор раздельны в том смысле, что ни одно прямое местоположение не является внутренним более чем одному объекту. Набор закрыт с точки зрения границ, что значит что каждый элемент в геометрическом комплексе имеет коллекцию (тоже геометрический комплекс) геометрических примитивов, которые представляют границу этого элемента. Таким образом, если наибольшее измерение геометрического примитива - 3D, композиция оператора границы в этом определении завершается после максимум трех шагов. Из этого также следует, что граница любого объекта - замкнута.

Геометрические комплексы часто называют чистой или внутренней топологией, что означает, что различные топологические несоответствия были убраны для получения «завершенности» представлений границ.
[ISO 19107]

4.14
геометрический объект (geometric object)
пространственный объект, представляющий геометрический набор

ПРИМЕЧАНИЕ: Геометричекий объект состоит из геометрического примитива, коллекции геометрических примитивов или геометрического комплекса рассматриваемого как единый объект. Геометрический объект может быть пространственным представлением другого объекта или его части. Независимо от представления, объект обычно считается топологически закрытым, так что считается, что точки границы объекта принадлежат этому объекту, при этом эти точки не всегда явно представлены в геометрическом объекте. При представлении топологического объекта, предполагается, что геометрические объекты не включают свои границы.
[ISO 19107]

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

4.15
геометрический примитив (geometric primitive)
геометрический объект (4.14) представляющий единичный, связный, однородный элемент пространства

ПРИМЕЧАНИЕ: Геометрические примитивы не разложен и представляют информацию о геометрической конфигурации. Они включают точки, кривые, поверхности и объемы. Несмотря на частое применение в обратном смысле, геометрические примитивы открыты и разлагаемы (могут быть разбиты на меньшие объекты) в связи с непрерывностью пространства. Примитивами являются объекты не подвергнутые такому разложению.
[ISO 19107]

4.16
внутреннее (interior)
набор **прямых местоположений** (4.7) принадлежащих **геометрическому объекту** (4.14) но не его **границе** (4.1)

ПРИМЕЧАНИЕ: Внутреннее топологического объекта – непрерывное изображение внутреннего любой геометрической реализации. Это не включено в определение, поскольку это следует из теоремы топологии. Другими словами, любая точка геометрического объекта принадлежит его внутреннему, если она может быть помещена внутрь гомеоморфного изображения открытого набора в Эвклидовом пространстве топологического измерения объекта.

4.17
линейная система координат (linear referencing system, linear positioning system)
система позиционирования, измеряющая расстояние от точки отсчета вдоль пути (объекта)

ПРИМЕЧАНИЕ: Система включает полный набор процедур по определению и получению записи для отдельных точек вдоль линейного объекта таких как метод(ы) определения точки, а также процедуры хранения, поддержки и получения информации о положении точек и сегментов маршрутов.
[NCHRP Synthesis 21, 1974]
[ISO 19133]

4.18
точка (point)
топологический 0-мерный **геометрический примитив** (4.15), представляющий положение

ПРИМЕЧАНИЕ: Граница точки – пустой набор.
[ISO 19107]

4.19
простой объект (simple feature)
объект, у которого все геометрические атрибуты описаны прямыми линиями или планарной интерполяцией между наборами точек

ПРИМЕЧАНИЕ: Интерполяция используется для кривых и поверхностей, которые по своей природе являются бесконечным набором точек и не подходят для конечного исчерпывающего представления. Каждый геометрический объект разлагается на части, которые могут быть выражены локально как параметрическая, линейная комбинация "контрольных точек." Это подробно описано в ISO 19107.

Для кривых, каждая часть ("сегмент" в ISO 19107) имеет две контрольные точки P0 (“начальная точка”) и P1 (“конечная точка”). Любая другая P на сегменте может быть описана, используя параметр *t* между 0.0 и 1.0 в “векторном” уравнении: $P = tP_0 + (1- t)P_1$.

Для поверхностей, каждая часть (“грань” в ISO 19107) может быть рассмотрена как полигон, который може быть на треугольники, каждый из которых состоит из трех контрольных точек P_0, P_1 и P_2. Каждая другая P треугольника может быть описана, используя 3 неотрицательных числа, чья сумма равна 1.0 (“барицентрические координаты”)

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

a,b,c ¡ $;a,b,c > 0;a + b + c = 1.0$ в векторном уравнении: $P = aP_0 + bP_1 + cP_2$.

4.20
начальная точка (start point)
first **point** (4.17) of a **curve** (4.7)
[ISO 19107]

4.21
поверхность (surface)
топологический 2-мерный **геометрический примитив** (4.15), локально представляющий нерерывное изображение района плоскости

ПРИМЕЧАНИЕ: Граница поверхности – набор ориентированных закрытых кривых очерчивающих границы поверхности.
[адаптировано из ISO 19107]

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

5 Символы и сокращения

5.1 Сокращения

API	Application Program Interface
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
DCE	Distributed Computing Environment
DCOM	Distributed Component Objected Model
DE-9IM	Dimensionally Extended Nine-Intersection Model
FID	Feature ID column in the implementation of feature tables based on predefined data types
GID	Geometry ID column in the implementation of feature tables based on predefined data types
IEEE	Institute of Electrical and Electronics Engineers, Inc.
MM	Multimedia
NDR	Little Endian byte order encoding
OLE	Object Linking and Embedding
RPC	Remote Procedure Call
SQL	Structured query language, not an acronym, pronounced as "sequel"
SQL/MM	SQL Multimedia and Application Packages
SRID	Spatial Reference System Identifier
SRTEXT	Spatial Reference System Well Known Text
UDT	User Defined Type
UML	Unified Modeling Language
WKB	Well-Known Binary (representation for example, geometry)
WKT	Well-Known Text
WKTR	Well-Known Text Representation
XDR	Big Endian byte order encoding

5.2 Символы

nD	n-мерный, n – любое целочисленное число
^{n}	n-мерное координатное пространство, n – любое целочисленное число
	пустой набор, набор не имеющий элементов
∩	пересечение, операция с двумя или более наборами
∪	объединение, операция с двумя или более наборами
-	разница, операция с двумя наборами
	принадлежит
	не принадлежит
	подмножество от, меньший набор не содержащий больший полностью
	подмножество от
	тогда и только тогла, логическая эквивалентность утверждений
	следовательно, логическое следствие где второе следует из первого
	существует
	для всех
	такое что
f: D → R	Функция "f" из домена "D" в диапазон "R"
{ X | s }	набор из "X" такой, что утверждение "s" ВЕРНО
	и, логическое пересечение
	или, логическое объединение
¬	не, логическое отрицание
=	равно
≠	не равно
	меньше или равно
<	меньше чем
	больше или равно

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

> больше чем
оператор топологической границы, соответствие геометрического объекта его границе

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

6 Архитектура

6.1 Объектная модель Geometry

6.1.1 Общее представление

В этом разделе рассматривается модель простых геометрических объектов[2]. Данная модель не привязана ни к какой Распределенной Вычислительной Платформе, а для её описания используется нотация UML. Схематичное изображение модели представлено на Рисунке 1. Базовый класс Geometry включает в себя подклассы Point, Curve, Surface и GeometryCollection. Каждый геометрический объект связан с Пространственной Системой Координат (Spatial Reference System), которая описывает координатное пространство, в котором определен данный геометрический объект.

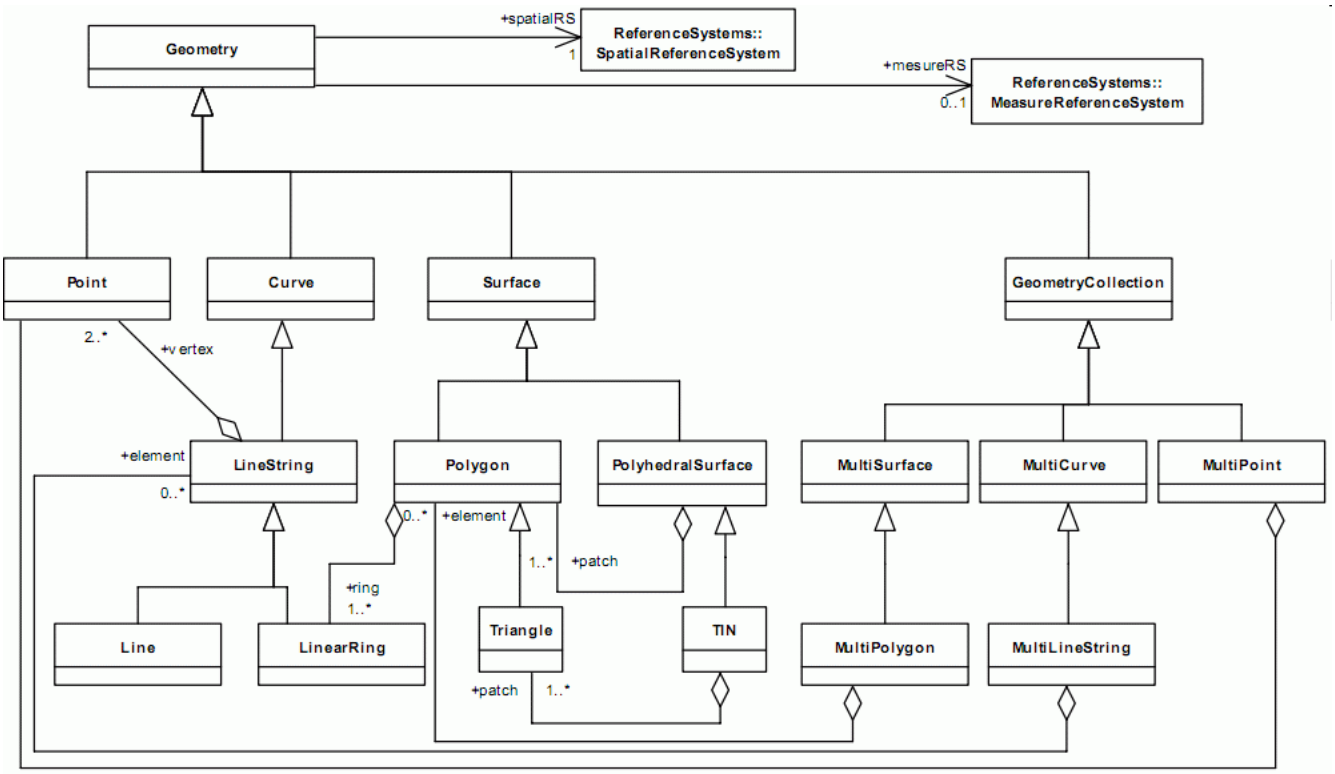


Рисунок 1: Иерархия класса Geometry

Рисунок 1 отражает расширенную геометрическую модель, включающую набор 0, 1 и 2-мерных классов – MultiPoint, MultiLineString и MultiPolygon, использующихся для описания коллекций Point, LineString и Polygon соответственно. MultiCurve и MultiSurface введены как суперклассы для обобщения интерфейсов управления Curve и Surface. Рисунок 1 показывает соединительные линии между классами, содержащими другие классы в качестве элементов и классами – элементами. Неоднородные коллекции являются экземплярами класса GeometryCollection. Свойства и методы каждого класса Geometry описаны ниже.

[2] http://en.wikipedia.org/wiki/Simple_Features

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

6.1.2 Geometry

6.1.2.1 Описание

Geometry - это корневой класс в иерархии классов. Geometry - это абстрактный класс, он не используется для создания экземпляров класса.

Классы-наследники Geometry могут иметь экземпляры класса и разделены на геометрические объекты размерности 0, 1 и 2, которые существуют в 2, 3 или 4-х мерном координатном пространстве (R^2, R^3, R^4). Геометрический объект в пространстве R^2 содержит точки с координатами x и y. Геометрические объекты в пространстве R^3 содержит координаты x, y и z или x, y и m. Объекты в пространстве R^4 содержат точки с координатами x, y, z и m. Интерпретация значений координат зависит от системы координат, в которой задана точка. Все точки объекта должны быть заданы в одной системе координат. Каждая координата должна быть однозначно связана с системой координат, которая задается для геометрического объекта.

Координата z обычно, но необязательно, представляет высоту, а координата m - некоторое измеренное значение.

Все подклассы Geometry в данной спецификации являются топологически замкнутыми, то есть описание их границ представлено набором точек. При других обстоятельствах возможно использование топологически открытых классов.

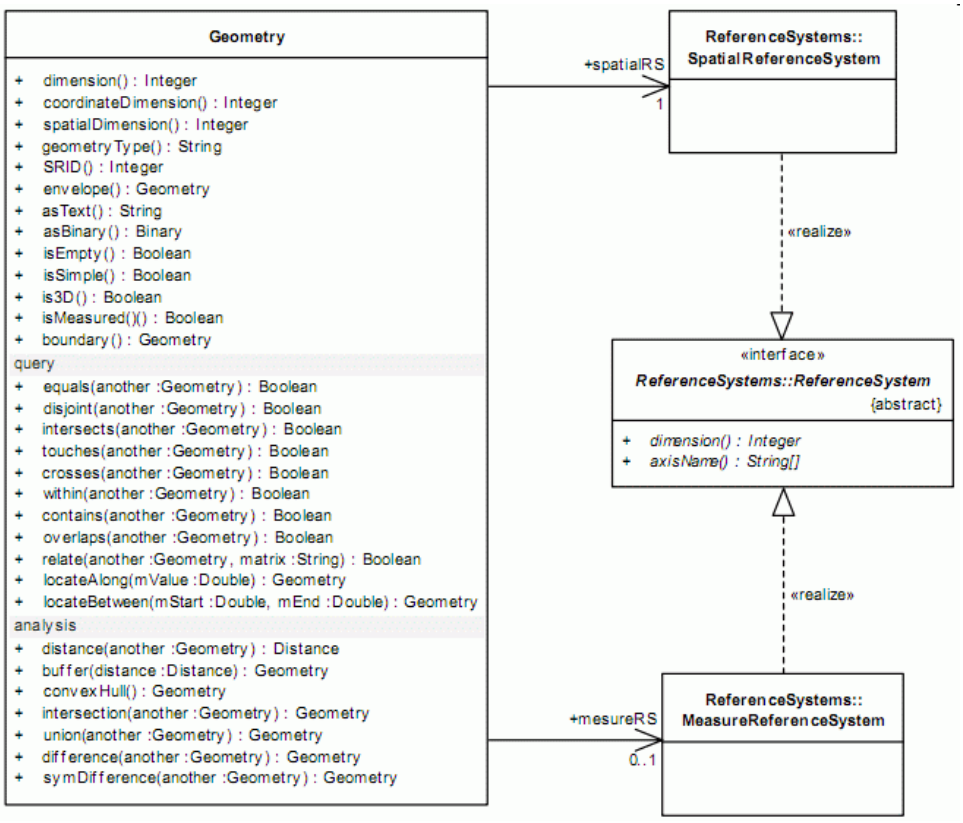


Рисунок 2: Операции класса Geometry

6.1.2.2 Базовые методы геометрических объектов

— **Dimension** (): Integer - возвращает измерение геометрического объекта, которое меньше или равно измерению координатного пространства. В неоднородных коллекциях данная функция будет возвращать максимальное измерение среди измерений, содержащихся в коллекции объектов.

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

— **GeometryType** (): String – возвращает имя создающего экземпляр подтипа для Geometry, на основе которого был создан данный экземпляр объекта. Имя подтипа Geometry возвращается как строка.

— **SRID** (): Integer **–** возвращает идентификатор пространственной системы координат (Spatial Reference System ID) геометрического объекта. Обычно это внешний ключ, ссылающийся на индекс таблицы в текущей либо в другой базе данных.

— **Envelope** (): Geometry **–** возвращает минимальный ограничивающий прямоугольник для данного объекта. Прямоугольник описывается координатами угловых точек [(MINX, MINY), (MAXX, MINY), (MAXX, MAXY), (MINX, MAXY), (MINX, MINY)]. Также могут быть добавлены минимальные/максимальные значения для Z и M. Простейшее представление Envelope это координаты двух точек, где одна содержит минимальные значения, вторая максимальные значения ограничивающего прямоугольника. В некоторых случаях эти координаты могут выходить за пределы корректных значений системы координат объекта.

— **AsText** (): String - возвращает текстовое представление объекта в формате WKT.

— **AsBinary** (): Binary **-** возвращает двоичное представление объекта в формате WKB.

— **IsEmpty** (): Integer – возвращает 1 (TRUE) если геометрический объект пустой, то есть представляет собой пустое множество точек Ø. Возвращаемое значение имеет тип integer (целое число), но интерпретируется как Boolean (логическое), TRUE=1, FALSE=0.

— **IsSimple** (): Integer – возвращает 1 (TRUE), если геометрический объект не пересекает и не касается сам себя. Каждый геометрический класс имеет свои определенные условия, которые классифицируют объект как НЕ простой. Возвращаемое значение имеет тип integer (целое число), но интерпретируется как Boolean (логическое), TRUE=1, FALSE=0.

— **Is3D** (): Integer – возвращает 1 (TRUE), если геометрический объект содержит z координату в списке координат точек.

— **IsMeasured** (): Integer – возвращает 1 (TRUE), если геометрический объект содержит m координату в списке координат точек.

— **Boundary** (): Geometry - возвращает геометрический объект, который описывает границу данного объекта. Возвращаемый объект является топологически замкнутым и представлен геометрическим примитивом (см. [1], раздел 3.12.2).

6.1.2.3 Методы определения пространственных отношений между геометрическими объектами

Методы данного раздела далее будут описаны более подробно для каждого геометрического класса. Для каждого метода возвращаемое значение имеет тип integer (целое число), но интерпретируется как Boolean (логическое), TRUE=1, FALSE=0.

— **Equals** (anotherGeometry: Geometry): Integer – возвращает 1 (TRUE), если данный геометрический объект совпадает с объектом anotherGeometry.

— **Disjoint** (anotherGeometry: Geometry): Integer – возвращает 1 (TRUE), если данный геометрический объект не пересекает объект anotherGeometry.

— **Intersects** (anotherGeometry: Geometry): Integer – возвращает 1 (TRUE), если данный геометрический объект пересекает объект anotherGeometry.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

— **Touches** (anotherGeometry: Geometry): Integer - возвращает 1 (TRUE), если данный геометрический объект касается объекта anotherGeometry.

— **Crosses** (anotherGeometry: Geometry): Integer - возвращает 1 (TRUE), если данный геометрический объект пересекает границу объекта anotherGeometry.

— **Within** (anotherGeometry: Geometry): Integer - возвращает 1 (TRUE), если данный геометрический объект находится внутри объекта anotherGeometry.

— **Contains** (anotherGeometry: Geometry): Integer – возвращает 1 (TRUE), если данный геометрический объект содержит объект anotherGeometry.

— **Overlaps** (anotherGeometry: Geometry): Integer – возвращает 1 (TRUE), если данный геометрический объект накрывает собой объект anotherGeometry.

— **Relate** (anotherGeometry: Geometry, intersectionPatternMatrix: String): Integer - возвращает 1 (TRUE), если данный геометрический объект соответствует топологическому отношению, заданному при помощи матрицы intersectionPatternMatrix, с объектом anotherGeometry. Метод проверяет пересечения между внутренними частями объектов (interior), границами объектов (boundary) и частями лежащими снаружи границы (exterior). Метод возвращает FALSE, если все проверяемые пересечения есть пустое множество, за исключением пересечения внешних частей объектов (exterior).

— **LocateAlong** (mValue: Double): Geometry - возвращает геометрический объект, содержащий все точки с совпадающей с mValue координатой m. Смотри раздел 6.1.2.6 «Измерения на геометрических объектах»

— **LocateBetween** (mStart: Double, mEnd: Double): Geometry - возвращает геометрический объект, содержащий все точки, координата m которых находится в диапазоне от mStart до mEnd включительно. Смотри раздел 6.1.2.6 «Измерения на геометрических объектах».

6.1.2.4 Методы пространственного анализа

Нижеследующие методы являются методами геометрического анализа. Результат их работы зависит от точности представления координат и ограничений, вызванных использованием в данной спецификации линейной интерполяции.

— **Distance** (anotherGeometry: Geometry): Double - возвращает кратчайшее расстояние между данным объектом и объектом anotherGeometry. Расстояние находится как расстояние между двумя ближайшими точками контуров объектов в системе координат данного объекта.

— **Buffer** (distance: Double): Geometry - возвращает геометрический объект, все точки которого находятся от данного объекта на расстоянии меньше или равно distance. Вычисления производятся в системе координат данного объекта. Вследствии ограничений линейной интерполяции часто возможна относительно небольшая ошибка в значении расстояния от контура буферного объекта до данного объекта, но она должна быть в пределах точности используемых координат.

— **ConvexHull** (): Geometry - возвращает геометрический объект, представляющий ограничивающий данный объект выпуклый многоугольник. Выпуклая оболочка вследствии использования линейной интерполяции должна состоять из прямых линий. Однако может быть точно представлена для любого объекта, также использующего линейную интерполяцию.

— **Intersection** (anotherGeometry: Geometry): - возвращает геометрический объект, содержащий набор точек фигуры, образованной пересечением данного объекта с объектом anotherGeometry.

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

— **Union** (anotherGeometry: Geometry): Geometry - возвращает геометрический объект, содержащий набор точек фигуры, образованной объединением данного объекта с объектом anotherGeometry.

— **Difference** (anotherGeometry: Geometry): Geometry - возвращает геометрический объект, содержащий набор точек фигуры, образованной вычитанием из данного объекта объекта anotherGeometry.

— **SymDifference** (anotherGeometry: Geometry): Geometry - возвращает геометрический объект, содержащий набор точек фигуры, образованной симметрической разностью данного объекта и объекта anotherGeometry.

6.1.2.5 Использование Z и M значений координат

Точка может содержать z координату. Координата z традиционно обозначает третье измерение (так называемое 3D). В геоинформационных системах (ГИС) это может быть высота над или под уровнем моря. Например, карта может содержать точку, обозначающую позицию горного пика на поверхности планеты при помощи значений x и y координат и высоту горы при помощи значения z координаты.

Кроме того, точка может содержать m координату. Значение m координаты позволяет приложению связать некоторые измерения с точками объекта. Например, сеть водостока может быть описана при помощи геометрического объекта мультиломаная с m координатой, хранящей расстояния от истока для каждой точки объекта. Метод LocateBetween может быть использован, чтобы найти части сети, которые расположены, например, между 10 и 12 километром от истока. Ограничений на содержимое m координаты нет, m координата не обязана содержать возрастающие значения вдоль контура объекта.

Методы определения положения объекта возвращают точки с z и m координатами, если данные координаты присутствуют. Пространственные операции выполняются в проекции карты и, следовательно, не используют z и m координаты в вычислениях (например функции Equals, Length) и при формировании новых геометрических объектов (например Buffer, ConvexHull, Intersection) также не используются z и m координаты. Данные операции выполняют проектирование геометрических объектов на горизонтальную плоскость, чтобы получить «отпечаток» или «тень» объекта. Другими словами, возможно хранить и получать z и m значения координат, но они не участвуют в пространственных операциях. Различные реализации могут включать истинные 3D пространственные операции, но должны оставаться совместимы со стандартом ISO 19107.

6.1.2.6 Измерения на геометрических объектах

Методы LocateAlong и LocateBetween вовзращают геометрический объект MultiPoint или MultiCurve, производные от исходного геометрического объекта, который содержит все координаты со значением m, попадающим в заданный диапазон. Метод LocateAlong является разновидностью метода LocateBetween. в котором начальное и конечное значение m совпадают (см. SQL/MM [1]).

6.1.2.6.1 Пустое множество

Для пустого множества возвращается значение null.

6.1.2.6.2 Геометрические объекты без m координаты

Для геометрического объекта без m координаты в списке точек возвращается пустое множество типа Point.

6.1.2.6.3 Геометрические объекты размерности 0

Для геометрических объектов нулевой размерности (точки) метод возвращает объекты Point или MultiPoint. Если подходящие условию координаты m не найдены, возвращается пустое множество типа Point.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Примеры:

а) Пусть метод LocateAlong вызывается с параметром m=4 для объекта, описывающегося в формате WKT следующим образом:
multipoint m(1 0 4, 1 1 1, 1 2 2, 3 1 4, 5 3 4)
тогда результатом работы метода будет объект:
multipoint m(1 0 4, 3 1 4, 5 3 4)

б) Пусть метод LocateBetween вызывается с начальным значением SM=2 и конечным значением EM=4 для объекта
multipoint m(1 0 4, 1 1 1, 1 2 2, 3 1 4, 5 3 5, 9 5 3, 7 6 7)
тогда результатом будет объект:
multipoint m(1 0 4, 1 2 2, 3 1 4, 9 5 3)

в) Пусть метод LocateBetween вызывается с начальным значением SM=1 и конечным значением EM=4 для объекта:
point m(7 6 7)
тогда результатом будет объект:
point m empty

г) Пусть метод LocateBetween вызывается с начальным значением SM=7 и конечным значением EM=7 для объекта:
point m(7 6 7)
тогда результатом будет объект:
multipoint m(7 6 7)

6.1.2.6.4 Геометрические объекты размерности 1

Для определения любой точки с нужной m координатой, лежащей на объекта размерности 1 в диапазоне между mStart и mEnd используется интерполяция. Алгоритм интерполяции может иметь различны реализации. Интерполяция используется внутри элемента Curve, но не между элементами объекта MultiCurve. Например, дано значение m=6 и объект LineString из 2-х точек, в котором первая точка имеет значение m=4 и вторая точка m=8. Таким образом, 6 это половина расстояний между 4 и 8. Алгоритм интерполяции даст точку, лежащую ровно посередине между первой и второй точками данной линии.

Результаты работы метода LocateBetween представлены коллекцией геометрических объектов. Если найдены точки, идущие последовательно с m координатой между mStart и mEnd включительно, тогда примитив-кривая должен быть добавлен к геометрической коллекции. Любые точки, идущие не подряд с m координатой между mStart и mEnd, также должны быть добавлены к геометрической коллекции. Если не найдены точки, с m координатой, удовлетворяющей условию, метод должен вернуть пустое множество типа ST_Point.

Примеры:

а) Пусть метод LocateAlong вызывается с параметром m=4 для объекта:
LineStringM(1 0 0, 3 1 4, 5 3 4, 5 5 1, 5 6 4, 7 8 4, 9 9 0)
тогда результатом будет объект:
MultiLineStringM((3 1 4, 5 3 4), (5 6 4, 7 8 4))

б) Пусть метод LocateBetween вызывается с начальным значением SM=2 и конечным значением EM=4 для объекта:
LineStringM(1 0 0, 1 1 1, 1 2 2, 3 1 3, 5 3 4, 9 5 5, 7 6 6)
тогда результатом будет объект:
MultiLineStringM((1 2 2, 3 1 3, 5 3 4))

в) Пусть метод LocateBetween вызывается с начальным значением SM=6 и конечным значением EM=9 для объекта:
LineStringM(1 0 0, 1 1 1, 1 2 2, 3 1 3, 5 3 4, 9 5 5, 7 6 6)

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

тогда результатом будет объект
MultiPointM(7 6 6)

г) Пусть метод LocateBetween вызывается с начальным значением SM=2 и конечным значением EM=4 для объекта:
MultiLineStringM((1 0 0, 1 1 1, 1 2 2, 3 1 3), (4 5 3, 5 3 4, 9 5 5, 7 6 6))
тогда результатом будет объект:
MultiLineStringM((1 2 2, 3 1 3),(4 5 3, 5 3 4))

д) Пусть метод LocateBetween вызывается с начальным значением SM=1 и конечным значением EM=3 для объекта:
LineStringM(0 0 0, 2 2 2, 4 4 4)
тогда результатом будет объект:
MultiLineStringM((1 1 1, 2 2 2, 3 3 3))

е) Пусть метод LocateBetween вызывается с начальным значением SM=7 и конечным значением EM=9 для объекта:
MultiLineStringM((1 0 0, 1 1 1, 1 2 2, 3 1 3), (4 5 3, 5 3 4, 9 5 5, 7 6 6))
тогда результатом будет объект:
PointM empty

6.1.2.6.5 Геометрические объекты размерности 2

Вычисления для геометрических объектов размерности 2 зависят от конкретной реализации.

6.1.3 GeometryCollection

6.1.3.1 Описание

GeometryCollection - это геометрический объект, содержащий несколько других геометрических объектов.

Все элементы GeometryCollection должны быть в одной системе координат, которая также является системой координат для GeometryCollection.

Других ограничений для элементов GeometryCollection нет. Классы-наследники GeometryCollection могут иметь ограничения на состав элементов. Это могут быть ограничения на размерность геометрических объектов или на степень пространственного перекрытия.

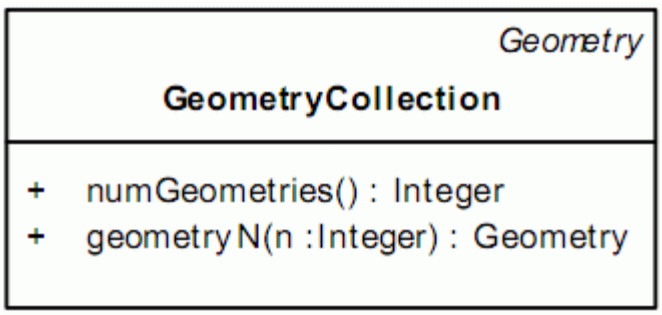


Рисунок 3: Операции GeometryCollection

6.1.3.2 Методы

Порядок элементов коллекции определяется способом их хранения. Две коллекции геометрических объектов, что отличаются только порядком элементов, считаются пространственно совпадающими и должны возвращать одинаковые результаты во всех геометрических операциях.

— **NumGeometries** ( ): Integer - возвращает количество элементов в данной коллекции.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

— **GeometryN** (N: integer): Geometry - возвращает N-й элемент коллекции.

6.1.4 Point

6.1.4.1 Описание

Point - это геометрический объект размерности 0, который обозначает единственную позицию в пространстве. Point содержит значения x и y координат. Координаты заданы в системе координат объекта. Point также может содержать значения z и m координат. Границы Point это всегда пустое множество.

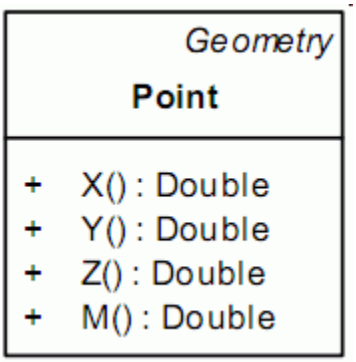


Рисунок 4: Point

6.1.4.2 Методы

— **X** ( ):Double — значение x координаты точки.
— **Y** ( ):Double — значение y координаты точки.
— **Z** ( ):Double — значение z координаты точки, если присутствует, иначе NIL.
— **M** ( ):Double — значение m координаты точки, если присутствует, иначе NIL.

6.1.5 MultiPoint

MultiPoint это GeometryCollection размерности 0. Элементами объекта MultiPoint могут являться только объекты Point, которые не соединены между собой и порядок их следования не важен (см. описание GeometryCollection). MultiPoint считается простым, если нет двух совпадающих Point в списке (Point с одинаковыми координатами x и y). Любой объект MultiPoint пространственно эквивалентен согласно определению в главе 6.1.15.3 простому объекту MultiPoint. Граница объекта MultiPoint есть пустое множество.

6.1.6 Curve

6.1.6.1 Описание

Curve - это геометрический объект размерности 1, обычно хранимый в виде последовательности Point с указанием типа интерполяции между ними. Данная спецификация определяет только один тип кривых – LineString, который использует линейную интерполяцию между точками.

Curve – это одномерный объект, гомеоморфичный реальному, closed, interval:

$$D = [a, b] = \{t \in \Re \mid a \leq t \leq b\}$$

после подстановки

$$f : [a, b] \rightarrow \Re^n$$

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

где n - размерность системы координат объекта.

Объект Curve является простым, если он не пересекает одну и ту же точку дважды. Исключением являются первая и последняя точки. (см. [1], раздел 3.12.7.3):

$$\forall\ c \in Curve,\ [a, b] = c.Domain,\ c =: f : [a, b] \rightarrow \Re^n$$
$$c.IsSimple \Leftrightarrow \forall\ x_1, x_2 \in [a, b]: [\ f(x_1)=f(x_2) \wedge x_1<x_2] \Rightarrow [x_1=a \wedge x_2=b]$$

Объект Curve считается замкнутым, если его первая и последняя точки совпадают. (см. [1], раздел 3.12.7.3).

$$c.IsClosed \Leftrightarrow [f(a) = f(b)]$$

Граница (boundary) замкнутого объекта Curve - есть пустое множество.

$$c.IsClosed \Leftrightarrow [c.boundary = \varnothing]$$

Замкнутый и простой Curve является Кольцом (Ring).

Границей незамкнутого объекта Curve являются две конечные точки.

Curve считается топологически замкнутым геометрическим объектом, следовательно он включает свои концевые точки f(a) и f(b).

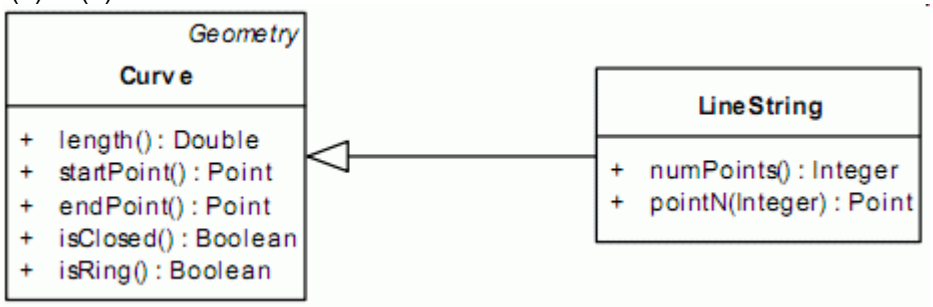


Рисунок 5: Curve

6.1.6.2 Методы

— **Length** ( ): Double - длина объекта Curve в его системе координат.

— **StartPoint** ( ): Point - первая точка Curve.

— **EndPoint** ( ): Point - последняя точка Curve.

— **IsClosed** ( ): Integer - возвращает 1 (TRUE), если [StartPoint() = EndPoint()].

— **IsRing** ( ): Integer - возвращает 1 (TRUE), если объект Curve замкнут и не проходит через одну и ту же точку дважды.

6.1.7 LineString, Line, LinearRing

6.1.7.1 Описание

LineString - это Curve с линейной интерполяцией между точками. Каждая последовательная пара точек определяет объект Line.

Line - это объект класса LineString, состоящий из 2-х точек.

LinearRing - это LineString, который одновременно простой и замкнутый. Curve на Рис. 6c есть замкнутый LineString, следовательно является LinearRing, Curve на Рис 6d также замкнут, но не является LinearRing.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

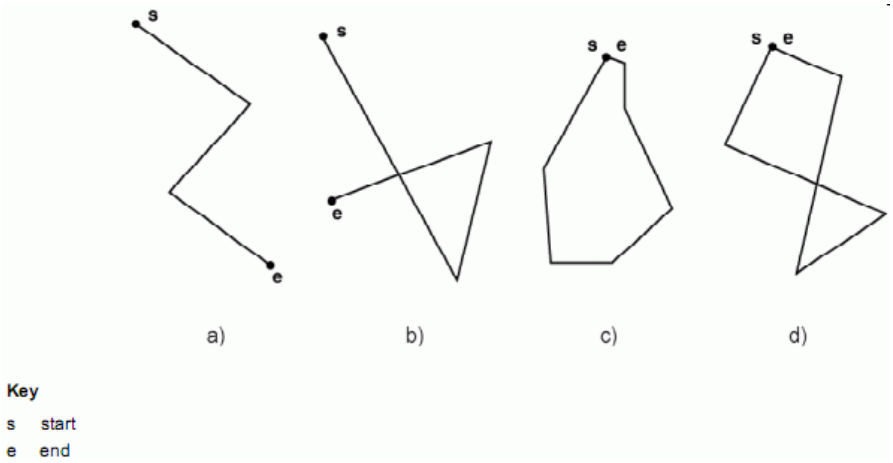


Рисунок 6: Примеры LineString
Простой LineString (a),
Непростой LineString (b),
Простой замкнутый LineString (LinearRing) (c),
Непростой замкнутый LineString (d)

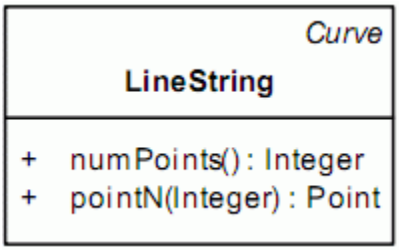


Рисунок 7: LineString

6.1.7.2 Методы

— **NumPoints** ( ): Integer - число точек в данном объекте LineString

— **PointN** (N: Integer): Point - возвращает указанную точку, принадлежащую LineString.

6.1.8 MultiCurve

6.1.8.1 Описание

MultiCurve это GeometryCollection размерности 1, элементами которой являются объекты Curve.

В данной спецификации MultiCurve не используется для создания объектов, он лишь определяет набор методов и включен для будущего расширения стандарта.

Объект MultiCurve считается простым, только если все его элементы являются простыми и пересекаются только в точках, являющихся конечными для обоих пересекающихся элементов.

Границы MultiCurve определяются по правилу объединения «mod 2». A Point is in the boundary of a MultiCurve if it is in the boundaries of an odd number of elements of the MultiCurve (Reference [1], section 3.12.3.2).

Объект MultiCurve является замкнутым, если все его элементы замкнутые. Граница замкнутого объекта MultiCurve - всегда пустое множество.

MultiCurve является топологически замкнутым.

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

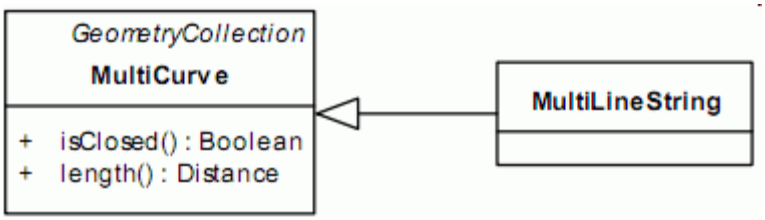


Рисунок 8: MultiCurve

6.1.8.2 Методы

— **IsClosed** ( ): Integer (Замкнут ли MultiCurve) - Возвращает 1 (TRUE), если для каждой кривой из коллекции StartPoint = EndPoint.

— **Length** ( ): Double (Длина) - Длина MultiCurve, которая определяется как сумма длин элементов.

6.1.9 MultiLineString

MultiLineString это объект MultiCurve, кривые в котором являются ломаными (LineString).

Границы MultiLineString на Рис. 9: (a) -{s1, e2}, (b) - {s1, e1}, (c) - Ø.

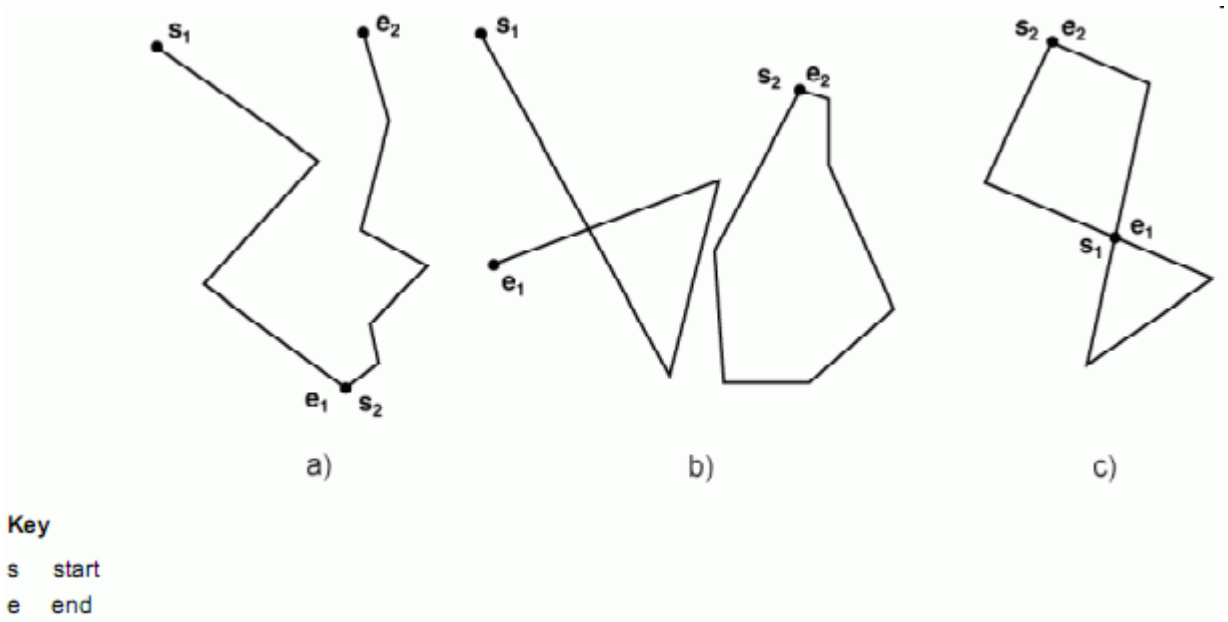


Рисунок 9: Примеры объектов MultiLineString

Примечание:

Рисунок вверху показывает: простой MultiLineString (a), непростой MultiLineString (b), непростой замкнутый MultiLineString, состоящий из 2-х элементов (c).

6.1.10 Surface

6.1.10.1 Описание

Surface - это геометрический объект размерности 2 (Поверхность).

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Простой объект Surface может состоять из одного фрагмента («patch»), который связан с одной внешней границей и 0 или более внутренними.

Многогранные (polyhedral) поверхности образованы путем соединения друг с другом простых фрагментов (patch) вдоль их общих границ. Такие поверхности в трехмерном пространстве могут быть не плоскими. Если все нормали к полигонам (фрагментам) параллельны друг к другу, тогда поверхность образованная данными полигонами является плоской и может быть представлена единым фрагментом.

Граница Surface представляется набором замкнутых объектов Curve, соответствующим внешним и внутренним границам полигонов, образующих поверхность. Создавать объекты можно от наследников класса Surface - Polygon и PolyhedralSurface. Полигон (Polygon) это простая плоская поверхность. Многогранная поверхность (PolyhedralSurface) это простая поверхность, содержащая некоторое число полигонов, которые являются фрагментами или гранями, составляющими поверхность. Если PolyhedralSurface закрыта, она ограничивает трехмерное тело. Мультиповерхность (MultiSurface), содержащая набор закрытых многогранных поверхностей, может быть использована для представления трехмерного объекта с дырками.

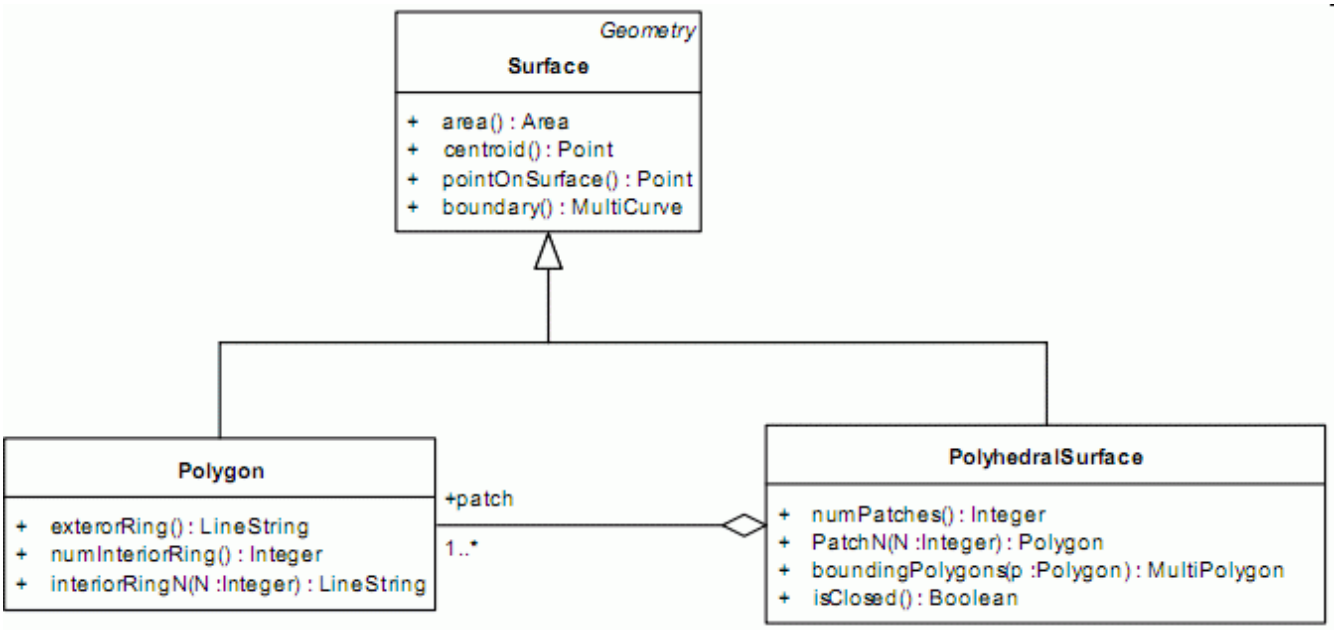


Рисунок 10: Surface

6.1.10.2 Методы

— **Area** ( ): Double - площадь Surface, измеренная в пространственной системе координат данного объекта.

— **Centroid** ( ): Point - математический центроид для Surface. Функция возвращает точку, которая может не находится на данной поверхности.

— **PointOnSurface** ( ): Point - функция возвращает точку, которая гарантированно принадлежит Surface

6.1.11 Polygon, Triangle

6.1.11.1 Описание

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Polygon (Полигон) это плоская поверхность, задаваемая при помощи одной внешней границы и 0 или более внутренних. Каждая внутренняя граница определяет дырку в полигоне. Triangle (Треугольник) это полигон с тремя несовпадающими и не лежащими на одной прямой вершинами. В треугольнике не может быть внутренних границ.

Внешние и внутренние границы задаются при помощи геометрического объекта LinearRing. Верхняя сторона поверхности - это сторона для которой внешние границы направлены против часовой стрелки. Внутренние границы при этом должны следовать по часовой стрелке.

Polygon должен удовлетворять следующим условиям:

а) Полигоны являются топологически замкнутыми объектами.
б) Граница полигона содержит набор объектов LinearRing, что образуют внешнюю и внутренние границы полигона.
в) Границы полигона не должны пересекаться. Пересечение допускается только в одной точке.

$$\forall\ P \in \text{Polygon},\ \forall\ c1,\ c2 \in P.\text{Boundary}(),\ c1 \neq c2,$$
$$\forall\ p,\ q \in \text{Point},\ p,\ q \in c1,\ p \neq q\ ,\ [\ p \in c2 \Rightarrow q \notin c2];$$

г) Полигон не должен содержать пересекающих его полилиний, шипов или проколов.

$$\forall\ P \in \text{Polygon},\ P = P.\text{Interior.Closure};$$

д) Внутренняя область полигона является связанным набором точек (можно соединить любые 2 точки во внутренней области не пересекая границы полигона).
е) Внешняя область полигона с дырками не является связанной (так как области, заданные дырками не принадлежат полигону и, соответственно, являются внешними).

В описанных условиях внешняя, внутренняя область и closure имеют стандартные топологические определения. Утверждения (а) и (в) определяют полигон как регулярный замкнутый набор точек. Полигоны - это простые геометрические объекты. Рисунок 11 представляет несколько примеров полигонов.

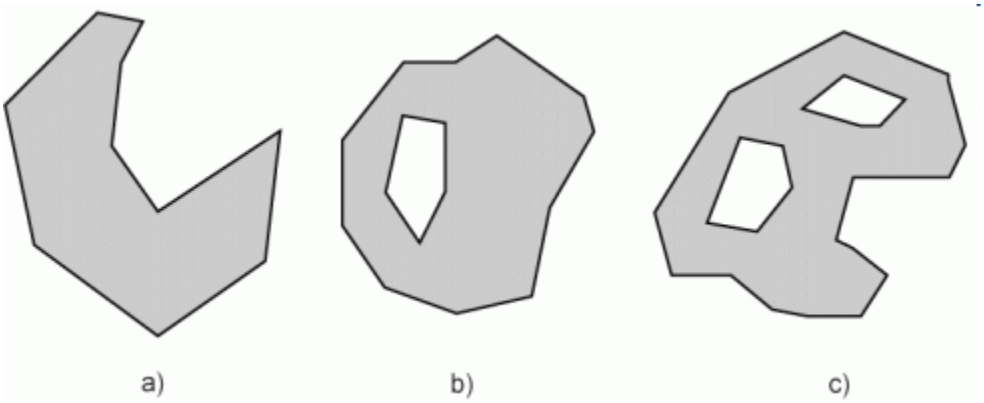


Рисунок 11: Примеры объектов Polygon
с одним (a), двумя (b) и тремя (c) кольцами (Ring)

Рисунок 12 показывает несколько примеров полигонов, которые не удовлетворяют описанным выше условиям и не могут быть представлены лишь одним полигоном.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

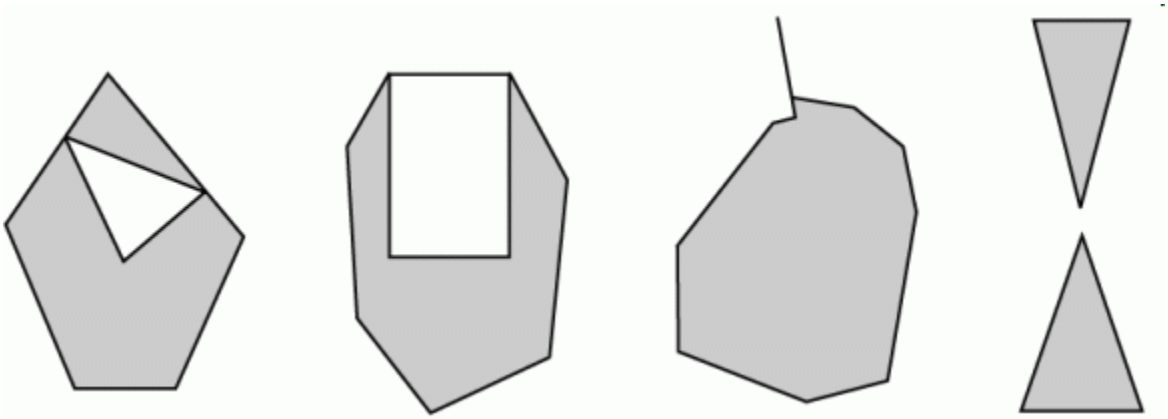


Рисунок 12: Примеры объектов, которые нельзя представить одним объектом Polygon

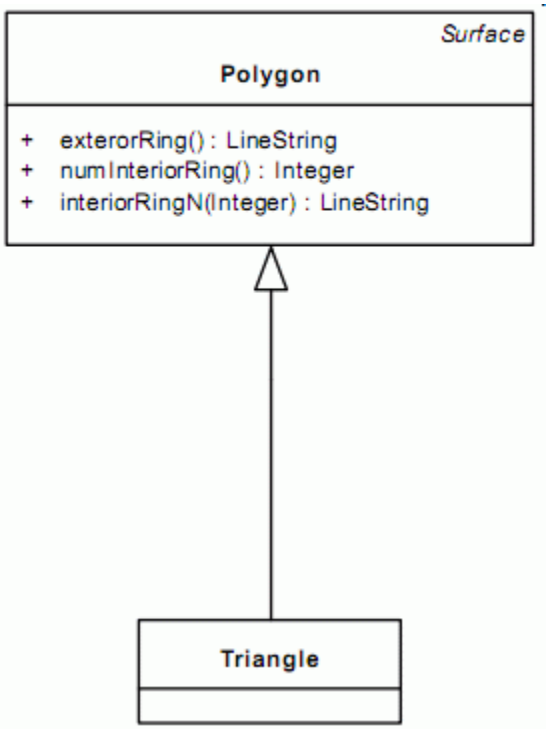


Рисунок 13: Polygon

6.1.11.2 Методы

— **ExteriorRing** ( ): LineString - возвращает внешнее кольцо объекта Polygon.

— **NumInteriorRing** ( ): Integer - возвращает число внутренних колец объекта Polygon.

— **InteriorRingN** (N: Integer): LineString - возвращает внутреннее N-е кольцо полигона.

6.1.12 PolyhedralSurface

6.1.12.1 Описание

Многогранная поверхность (PolyhedralSurface) является набором граничащих по внешней границе полигонов. Для каждой пары соседних полигонов общая граница выражается конечным набором

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

ломаных (LineStrings). Каждая такая ломаная является частью границы не более 2-х полигонов. TIN (нерегулярная сеть треугольников) это PolyhedralSurface, состоящая только из треугольников.

Все полигоны поверхности должны иметь одинаковое направление обхода. Это означает, что когда две полилинии двух соседних полигонов пересекают общую границу, они делают это в противоположных направлениях. Так как PolyhedralSurface неразрывна, все ее полигоны будут ориентированы в одном направлении. Это означает, что не ориентируемая поверхность (такая как лента Мебиуса) не может быть представлена одной поверхностью PolyhedralSurface, ее можно представить как Мультиповерхность (MultiSurface). Рисунок 14 показывает пример ориентированной поверхности (вид сверху). Стрелки показывают направление колец (LinearRings) из границ полигонов, в которых расположены стрелки.

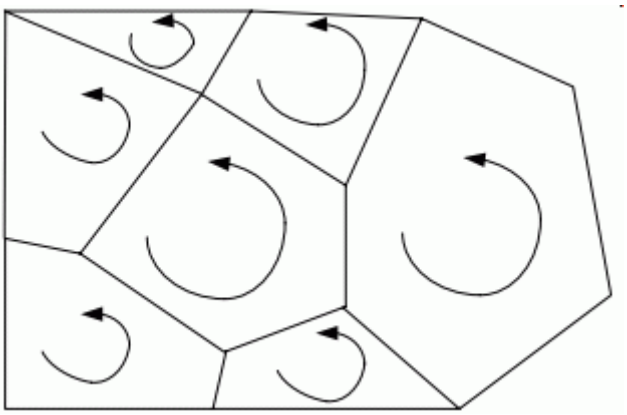


Рисунок 14: Polyhedar Surface с одинаковой ориентацией полигонов

Если каждая ломаная (LineString) является границей ровно 2-х полигонов, тогда PolyhedralSurface является простым, замкнутым полигидроном (polyhedron) и топологически изоморфен к поверхности сферы. Согласно теоремы Джордана о 2-х сферах (Jordan's Theorem for 2-spheres) данный полигидрон образует трехмерное тело топологически изоморфное к внутренней части сферы, в данном случае шар. Верхняя сторона поверхности будет снаружи или изнутри этого тела. Если снаружи, тогда поверхность образует внешнюю границу трехмерного тела, иначе внутреннюю. Шар с пустотами (дырками) внутри может быть представлен как одна внешняя оболочка и нужное количество внутренних оболочек.

6.1.12.2 Методы

— **NumPatches** () : Integer - возвращает число полигонов, составляющих поверхность.

— **PatchN** (N: Integer): Polygon - возвращает полигон N поверхности, порядок полигонов произвольный.

— **BoundingPolygons** (p: Polygon): MultiPolygon - возвращает набор полигонов данной поверхности, что ограничивают указанный полигон p, для любого полигона p поверхности.

— **IsClosed** (): Integer - возвращает 1 (TRUE), если поверхность замкнута и не имеет границы, образуя таким образом трехмерное тело.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

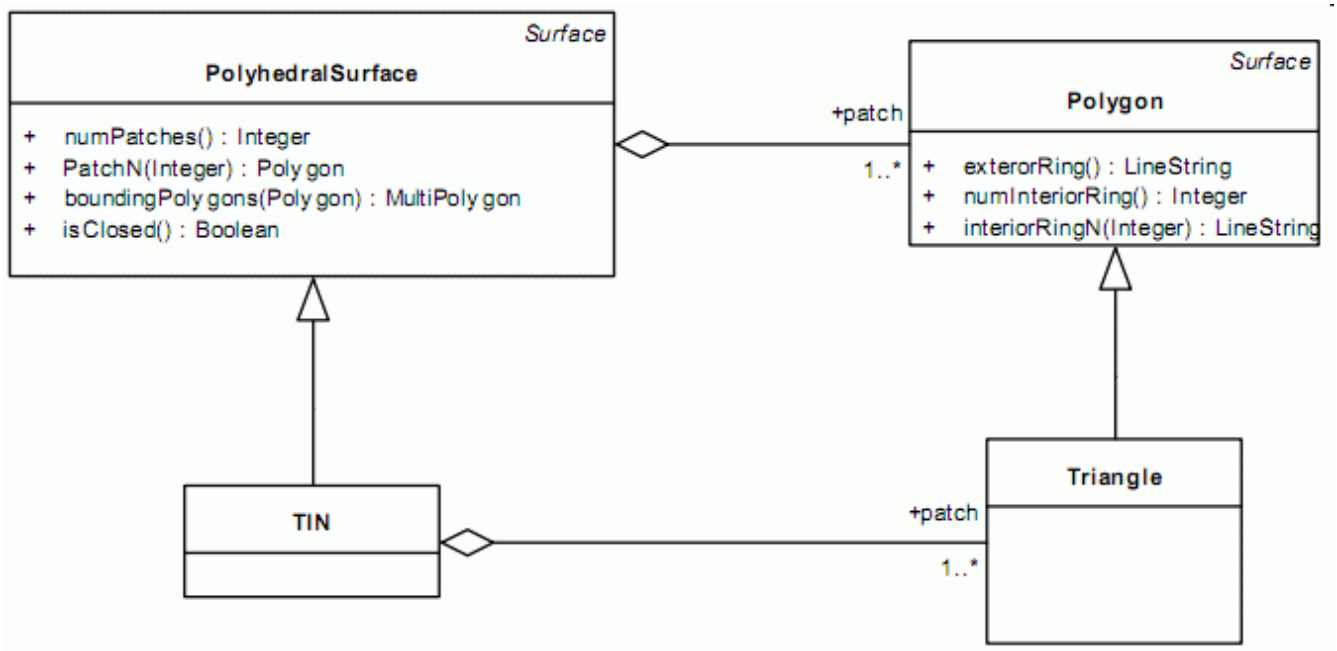


Рисунок 15: PolyhedralSurface

6.1.13 MultiSurface

6.1.13.1 Описание

MultiSurface это GeometryCollection размерности 2, элементами которой являются объекты Surface (Мультиповерхность). Все объекты коллекции должны использовать одну пространственную систему координат. Внутренние части поверхностей не должны пересекаться между собой. Границы двух копланарных поверхностей могут пересекаться в конечном числе точек. Если это случается вдоль кривой, поверхности могут быть объединены в одну.

MultiSurface это не абстрактный класс в данной спецификации и может быть использован для представления коллекций полигонов и многогранных поверхностей. Класс определяет набор методов для всех наследников. Наследником MultiSurface является MultiPolygon (Мультиполигон), элементами которого являются только полигоны. Если элементы коллекции не являются только полигонами, то следует использовать класс MultiSurface.

Примечание:
Геометрические отношения задаются в общей системе координат. Двумерные операции над картой, описанные в главе 6.1.15 могут классифицировать элементы как трехмерные MultiSurface, имеющие перекрывающиеся внутренние части в своих проекциях на плоскость.

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

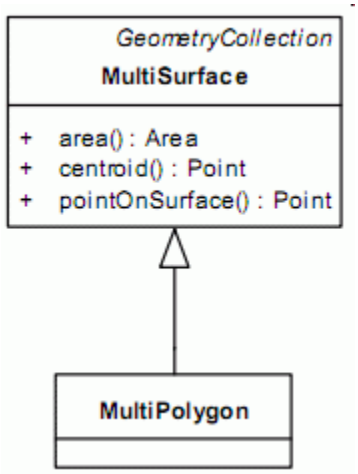


Рисунок 16: Операции MultiSurface

6.1.13.2 Методы

MultiSurface наследует методы NumGeometries и GeometryN от своего предка GeometryCollection для доступа к элементам коллекции.

— **Area** ( ): Double - площадь MultiSurface, измеренная в своей пространственной системе координат.

— **Centroid** ( ): Point - математический центроид MultiSurface. Точка не гарантированно находится на данном объекте.

— **PointOnSurface** ( ): Point - точка, гарантированно лежащая на объекте MultiSurface.

6.1.14 MultiPolygon

MultiPolygon (Мультиполигон) - это MultiSurface, элементами которой являются полигоны.

Для объекта MultiPolygon действуют следующие утверждения:

а) Внутренние области полигонов, являющиеся элементами MultiPolygon, не должны пересекаться.

$$\forall M \in MultiPolygon,\ \forall P_i, P_j \in M.Geometries(), i \neq j,$$
$$Interior(P_i) \cap Interior(P_j) = \emptyset;$$

б) Границы любых 2-х полигонов из MultiPolygon не могут пересекаться, а могут лишь касаться в конечном числе точек.

$$\forall M \in MultiPolygon,\ \forall P_i, P_j \in M.Geometries(),$$
$$\forall c_i, c_j \in Curve\ c_i \in P_i.Boundaries(),\ c_j \in P_j.Boundaries()$$
$$\exists k \in Integer\ \ni\ c_i \cap c_j = \{p_1, \ldots, p_k \mid p_m \in Point,\ 0<m<k\};$$

ПРИМЕЧАНИЕ: Crossing is prevented by assertion (a) above.

в) MultiPolygon является топологически замкнутым.

г) MultiPolygon не может иметь пересекающих линий, шипов или проколов, Мультиполигон это регулярный замкнутый набор точек.

$$\forall\ M \in MultiPolygon,\ M = Closure(Interior(M))$$

д) Внутренняя область MultiPolygon с более чем одним полигоном не является связанной. Количество связанных областей MultiPolygon соответствует количеству полигонов.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Граница MultiPolygon является набором замкнутых объектов Curve (LineString), которые в свою очередь являются границами полигонов. Каждая кривая в границе MultiPolygon является границей только одного полигона, и каждая кривая в границе полигона является частью границы MultiPolygon.

Читатель может ознакомится с работами Worboys ([13], [14]) и Clementini ([5], [6]) для понимания определения объекта MultiPolygon.

Рисунок 17 показывает 4 примера правильных объектов MultiPolygon с 1, 3, 2 и 2-мя полигонами соответственно.

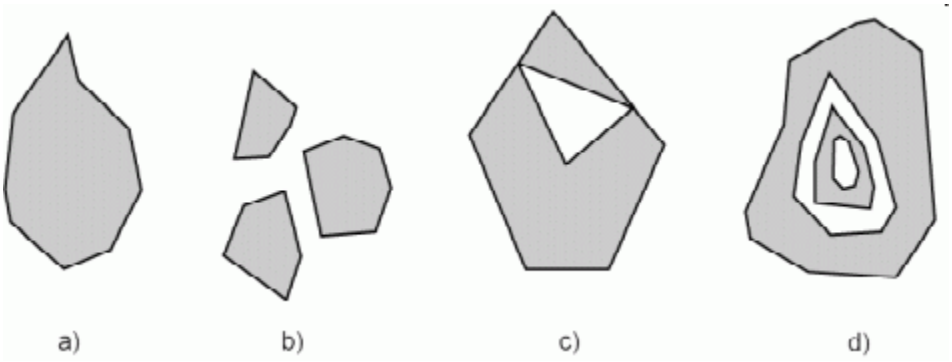


Рисунок 17: Примеры объектов MultiPolygon с 1 (a), 3(b), 2(c) и 2(d) полигонами

Рисунок 18 показывает примеры геометрических объектов, которые не могут быть представлены одним объектом MultiPolygon.

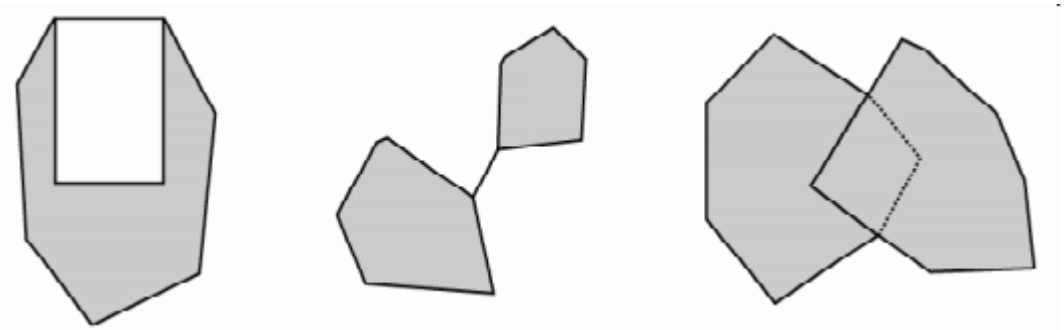


Рисунок 18: Примеры геометрических объектов, которые не могут быть представлены одним объектом MultiPolygon

Примечание:
Наследник класса Surface названный PolyhedralSurface как описано в [1] является поверхностью, состоящей из граней, гранями которой являются полигоны. PolyhedralSurface не является MultiPolygon, так как не удовлетворяет описанному выше правилу, в котором говорится, что границы полигонов могут пересекаться только в ограниченном числе точек (не допускается пересечение отрезков, лежащих на одной прямой[3]).

6.1.15 Операции отношений

6.1.15.1 Введение

[3] Примечание переводчика

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Операции отношения это логические методы, которые используются для определения существования заданных пространственных топологических отношений между двумя геометрическими объектами, в том виде, в котором они будут представлены на карте. Топологические пространственные отношения между двумя геометрическими объектами были изучены в [4], [5], [6], [7], [8], [9], и [10]. Основной подход в сравнении двух геометрических объектов состоит в проектировании их в плоскую горизонтальную систему координат, представляющую земную поверхность, и выполнении проверок на пересечение между внутренними областями проекций объектов, их границами и внешними областями. Затем пространственные отношения классифицируются при помощи полученной матрицы «пересечений», размерности 3 на 3. Внутренняя область, граница и внешняя область задается набором точек и описана в общей топологии, см [4].

Важно понимать, что вычисления следующих операций дадут эквивалентные результаты, используя классическое геометрическое представление и используя алгебраические методы, оперирующие с эквивалентными топологическими структурами.

Описанные принципы заложены в данном стандарте для определения пространственных отношений между двумерными геометрическими объектами в двумерном пространстве. Объекты проецируются на горизонтальную поверхность, обычно представляющую карту. Данный подход даст различные результаты, если сравнить его с полной трехмерной геометрией (и соответствующей ей трехмерной топологией), из-за изменений, внесенных в объекты при проецировании их на поверхность. Можно определить полный набор трехмерных пространственных операций, но это увеличит сложность вычислений и может стать препятствием для большинства реализаций стандарта. Кроме того трехмерные топологические операции не поддерживаются многими геоинформационными системами работающими с большими объемами картографических данных. Спецификация трехмерных топологических операций зарезервирована для будущих версий этого стандарта.

Примечание: Важно помнить, что когда в данном стандарте говорится о взаимоотношениях, лежащих в основе операций отношений, имеются ввиду полные взаимоотношения в пространственной системе координат объектов, участвующих в них, за исключением случаев, когда это будет оговорено отдельно.

Для применения концепции внутренней области, границы и внешней области в отношении объектов размерности 0 и 1 в двумерном пространстве будет применен подход, называемый комбинированной топологией (см [1] раздел 3.12.3.2). Данный подход основан на принятых определениях границ, внутренних и внешних областей (см [12]) и ведет к следующим результатам.

Граница геометрического объекта - это набор геометрических объектов размерности на единицу меньше размерности исходного объекта. Границы Point или MultiPoint - пустое множество. Граница незамкнутой кивой Curve состоит из двух её конечных точек, граница замкнутой кривой - пустое множество. Граница объекта MultiCurve содержит точки, that are in the boundaries of an odd number of its element Curves. Граница объекта Polygon содержит набор его Колец (Rings). Граница объекта MultiPolygon содержит набор Rings, являющиеся границами полигонов - элементов коллекции. Граница произвольной коллекции геометрических объектов, внутренние области которых не соединяются. содержит геометрические объекты из границ элементов коллекции, отобранных по правилу объединения "mod 2" (см. [1] раздел 3.12.3.2).

Описанные области геометрических объектов считаются изолированными. Внутренняя область геометрического объекта содержит точки, которые покинут ее, если граница будет убрана. Внешняя область геометрического объекта содержит точки, которые отсутствуют и в границе и во внутренней области.

Изучения взаимоотношений геометрических объектов максимальной размерности 1 или 2, предполагавшие проверки на пересечения внутренних областей и границ двух объектов, привели к определению модели 4-х пересечений (см. [8]). Данная модель была расширена до модели 9-ти пересечений путем добавления внешних областей проверяемых объектов (см [11]). И в дальнейшем в нее была включена информация о размерности объектов-результатов пересечений. Данная модель называется Модель 9-ти пересечений с указанием размерности (см [5]). Последнее усовершенствование позволило данной модели описывать пространственные отношения между точками, линиями и

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

полигонами, включая полигоны с дырками и полигоны или линии, состоящие из нескольких частей (см [6]).

6.1.15.2 Модель 9-ти пересечений с указанием размерности

Дан геометрический объект a, пусть I(a), B(a) и E(a) обозначают внутреннюю область, границу и внешнюю область объекта a, соответственно.

Пусть dim(x) показывает максимальную размерность (-1, 0, 1 или 2) геометрических объектов x, где значение -1 обозначает dim(Ø).

Пересечение любых двух I(a), B(a) или E(a) могут образовать множество геометрических объектов x, состоящее из объектов различной размерности. Например, пересечение границ двух полигонов может содержать точку и линию.

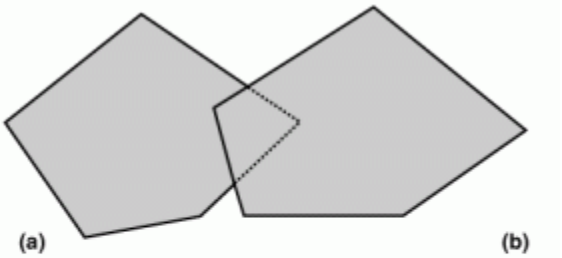
Таблица 1 показывает общую форму матрицы модели 9-ти пересечений с указанием размерности (DE-9IM).

Таблица 1: DE-9IM

	Interior	Boundary	Exterior
Interior	$dim(I(a) \cap I(b))$	$dim(I(a) \cap B(b))$	$dim(I(a) \cap E(b))$
Boundary	$dim(B(a) \cap I(b))$	$dim(B(a) \cap B(b))$	$dim(B(a) \cap E(b))$
Exterior	$dim(E(a) \cap I(b))$	$dim(E(a) \cap B(b))$	$dim(E(a) \cap E(b))$

Для регулярных, топологически замкнутых геометрических объектов вычисление максимальной размерности пересечения внутренней, внешней областей и границ не требует явного вычисления и представления результирующих множеств, полученных при пересечении. Для вычисления пересекаются ли внутренние области двух полигонов и для определения размерности образованного множества нет необходимости явно представлять внутренние области полигонов, как отдельные геометрические объекты. В большинстве случаев размерность результирующего множества в ячейке таблицы имеет ограниченный возможный набор значений, зависящий от типа пересекающихся геометрических объектов. В случае пересечения линии и полигона в ячейке пересечения внутренних областей допустимы значения размерности от -1 до 1. В случае пересечения двух полигонов, в ячейке результата пересечения внутренних областей допустимы значения от -1 до 2. В таких случаях кроме определения факта пересечения никакого дополнительного анализа не требуется.

Рисунок 19 показывает пример DE-9IM для случая, когда a и b являются перекрывающиеся объекты Polygon.



	Interior	Boundary	Exterior
Interior	2	1	2
Boundary	1	0	1
Exterior	2	1	2

Рисунок 19: Пример пересечения и соответствующая DE-9IM

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Для двух геометрических объектов предикат пространственного отношения может быть выражен как формула, на вход которой подается матрица, представляющая собой множество допустимых значений DE-9IM для двух геометрических объектов. Если пространственные отношения между двумя геометрическими объектами соответствуют значениям входной матрицы, тогда предикат возвращает TRUE.

Входная матрица является образцом и содержит 9 значений, по одному для каждой ячейки таблицы пересечений. Допустимые значения p матрицы являются {T, F, *, 0, 1, 2} и их определение для любой ячейки таблицы, где x представляет множество объектов образованных при пересечении, дано ниже:

```
p = T ⇒ dim(x) ∈ {0, 1, 2}, i.e. x ≠ ∅
p = F ⇒ dim(x) = -1, i.e. x = ∅
p = * ⇒ dim(x) ∈ {-1, 0, 1, 2}, i.e. Don't Care
p = 0 ⇒ dim(x) = 0
p = 1 ⇒ dim(x) = 1
p = 2 ⇒ dim(x) = 2
```

Матрица-образец может быть представлена как массив или список из 9-ти символов, представленный в виде строки, ячейки таблицы пересечений в которой представлены построчно. Например, следующий фрагмент кода может быть использован для проверки наложения двух областей:

```
char * overlapMatrix = "T*T***T**";
Geometry* a, b;
Boolean b = a->Relate(b, overlapMatrix);
```

6.1.15.3 Именнованные предикаты для проверки пространственных отношений, основанные на DE-9IM

Предикат Relate (Относится), основанный на входной матрице-образце имеет преимущество в том, что пользователь может выполнить проверку для большого числа пространственных отношений и может быть настроен для проверки любого особенного отношения между тестируемыми объектами. Недостатком данного предиката является то, что он представляет собой низкоуровневую функцию и не имеет аналога в естественном языке. Пользователями геопространственной системы являются IT специалисты, использующие COM API в языках программирования, таких как Visual Basic, пользователи SQL, которые желают, например, отобрать все объекты попадающие внутрь полигона, а также более искушенные разработчики ГИС.

Чтобы удовлетворить запросы таких пользователей для модели DE-9IM был разработан список именованных предикатов (см [5,6]). Пять предикатов имеют названия: Disjoint(Не пересекается), Touches (Касается), Crosses (Пересекает), Within (Внутри) и Overlaps (Накладывается). Определения данных предикатов даны ниже. В этих определениях символ P используется для задания геометрических объектов размерности 0 (Точки (Points) и Мультиточки (MultiPoints)), L используется для задания объектов размерности 1 (Ломаной (LineStrings) и Мультиломаной (MultiLineStrings)) и A используется для задания объектов размерности 2 (Полигоны(Polygons) и Мультиполигоны (MultiPolygons)).

Equals

Даны два (топологически замкнутые) геометрических объекта "a" и "b":

$$a.Equals(b) \Leftrightarrow a \subseteq b \wedge b \subseteq a$$

Выражаясь в терминах DE-9IM:

$$\begin{aligned} a.Equals(b) \Leftrightarrow [\quad & (I(a) \cap I(b) \neq \emptyset) \wedge \\ & (I(a) \cap B(b) = \emptyset) \wedge \\ & (I(a) \cap E(b) = \emptyset) \wedge \end{aligned}$$

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
        (B(a) ∩ I(b) = ∅) ∧
        (B(a) ∩ B(b) ≠ ∅) ∧
        (B(a) ∩ E(b) = ∅) ∧
        (E(a) ∩ I(b) = ∅) ^
        (E(a) ∩ B(b) = ∅) ^
        (E(a) ∩ E(b) ≠ ∅) ]
⇔  a.Relate(b, "TFFFTFFFT")
```

Disjoint

Даны два (топологически замкнутые) геометрических объекта "a" и "b":

```
a.Disjoint(b) ⇔ a ∩ b = ∅
```

Выражаясь в терминах DE-9IM:

```
a.Disjoint(b) ⇔ [  (I(a) ∩ I(b) = ∅) ∧
                   (I(a) ∩ B(b) = ∅) ∧
                   (B(a) ∩ I(b) = ∅) ∧
                   (B(a) ∩ B(b) = ∅) ]
              ⇔  a.Relate(b, "FF*FF****")
```

Touches

Отношение Touches между двумя геометричесикми объектами "a" и "b" применимо к парам объектов типа A/A, L/L, L/A, P/A и P/L, но не к P/P. Оно определяется как:

```
a.Touch(b) ⇔ (I(a)∩I(b)=∅)∧(a∩b)≠∅
```

Выражаясь в терминах DE-9IM:

```
[a.Touch(b) ⇔ [    (I(a) ∩ I(b) = ∅) ∧
                   [  (B(a) ∩ I(b) ≠ ∅) ∨
                      (I(a) ∩ B(b) ≠ ∅) ∨
                      (B(a) ∩ B(b) ≠ ∅) ] ]
              ⇔ [  a.Relate(b, "FT*******") ∨
                   a.Relate(b, "F**T*****") ∨
                   a.Relate(b, "F***T****") ]
```

Рисунок 20 показывает несколько примеров отношения Touches.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

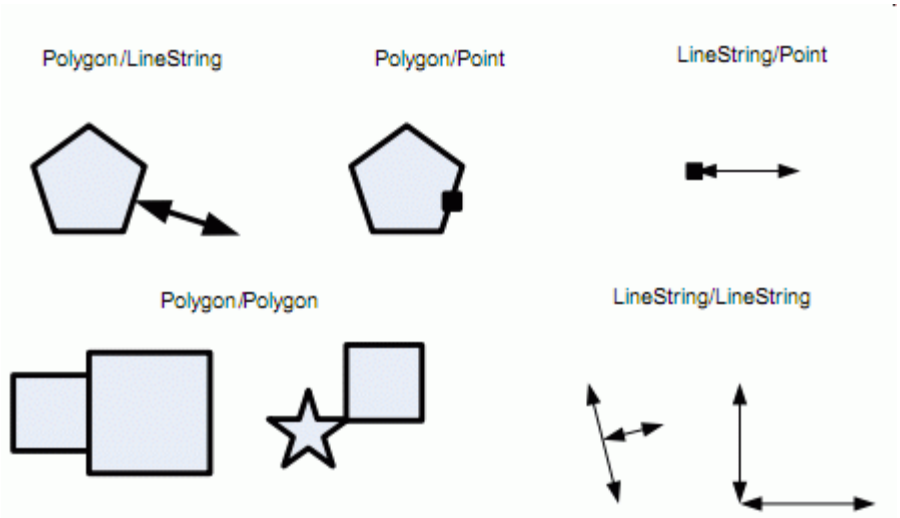


Рисунок 20: Примеры отношения Touches

Crosses

Отношение Crosses применимо к ситуациям, когда проверяемые объекты имеют типы P/L, P/A, L/L и L/A. Оно определяется как:

```
a.Cross(b) ⇔ [ dim(I(a)∩I(b))< max(dim(I(a)), dim(I(b)) ∧
                    (a ∩ b ≠a) ∧ (a ∩ b ≠b)]
```

Выражаясь в терминах DE-9IM:

```
Case a∈P, b∈L or
Case a∈P, b∈A or
Case a∈L, b∈A:
  a.Cross(b) ⇔ [ I(a)∩I(b)≠∅ ∧ I(a)∩E(b)≠∅ ]
               ⇔ a.Relate(b, "T*T******")

Case a∈L, b∈L:
  a.Cross(b) ⇔ dim(I(a)∩I(b)) = 0
                  ⇔ a.Relate(b, "0********");
```

Рисунок 21 показывает несколько примеров отношения Crosses.

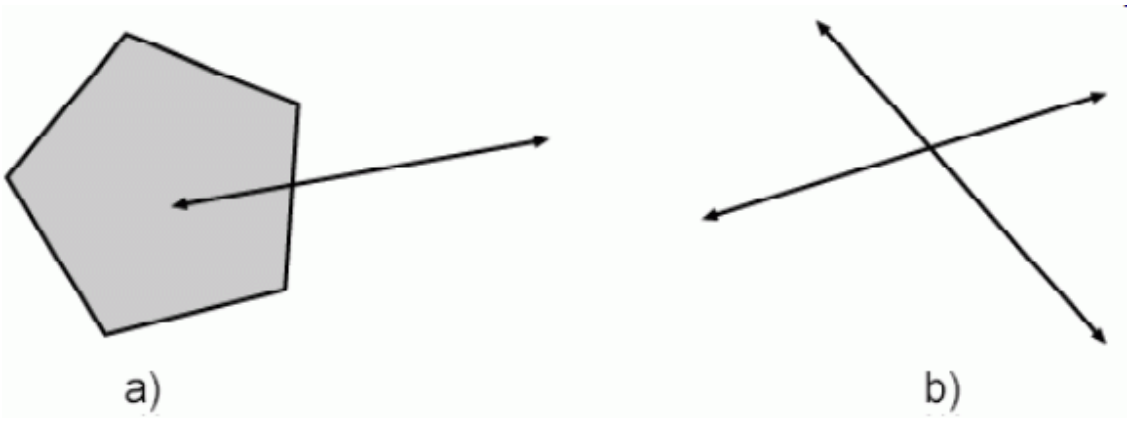


Рисунок 21: Примеры отношения Crosses
Polygon/LineString(a) и LineString/LineString(b)

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Within

Отношение Within определяется как:

$$a.Within(b) \Leftrightarrow (a \cap b = a) \wedge (I(a) \cap E(b) = \varnothing)$$

Выражаясь в терминах DE-9IM:

$$a.Within(b) \Leftrightarrow [\ I(a) \cap I(b) \neq \varnothing \wedge I(a) \cap E(b) = \varnothing \wedge B(a) \cap E(b) = \varnothing\ ]$$
$$\Leftrightarrow a.Relate(b, \text{“}T*F**F***\text{”})$$

Рисунок 22 показывает несколько примеров отношения Within.

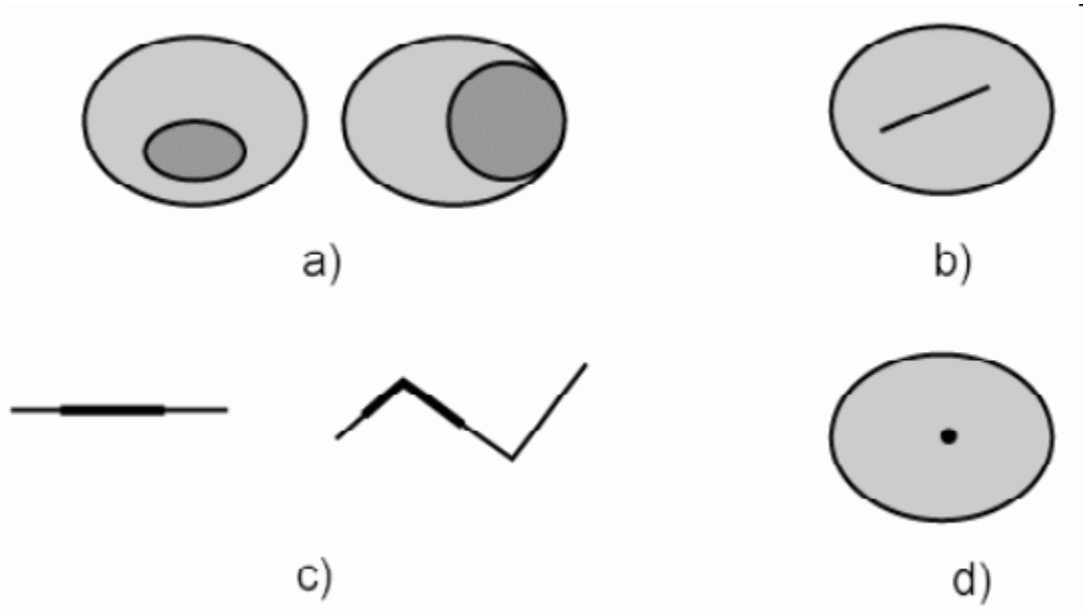


Рисунок 22: Примеры отношения Within
Polygon/Polygon(a), Polygon/LineString(b), LineString/LineString(c) и Polygon/Point(d)

Overlaps

Отношение Overlaps применимо к ситуациям, когда проверяемые объекты имеют типы A/A, L/L и P/P. Оно определяется как:

$$a.Overlaps(b) \Leftrightarrow (\ dim(I(a)) = dim(I(b)) = dim(I(a) \cap I(b)))$$
$$\wedge (a \cap b \neq a) \wedge (a \cap b \neq b)$$

Выражаясь в терминах DE-9IM:

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
Case a ∈ P, b ∈ P or Case a ∈ A, b ∈ A:

   a.Overlaps(b)  ⇔     (I(a) ∩ I(b) ≠ ∅) ∧

                        (I(a) ∩ E(b) ≠ ∅) ∧

                        (E(a) ∩ I(b) ≠ ∅)

                  ⇔ a.Relate(b, "T*T***T**")

Case a ∈ L, b ∈ L:

   a.Overlaps(b) ⇔ (dim(I(a) ∩ I(b) = 1) ∧ (I(a) ∩ E(b) ≠ ∅) ∧ (E(a) ∩ I(b) ≠
      ∅) ⇔ a.Relate(b, "1*T***T**")
```

Рисунок 23 показывает несколько примеров отношения Overlaps.

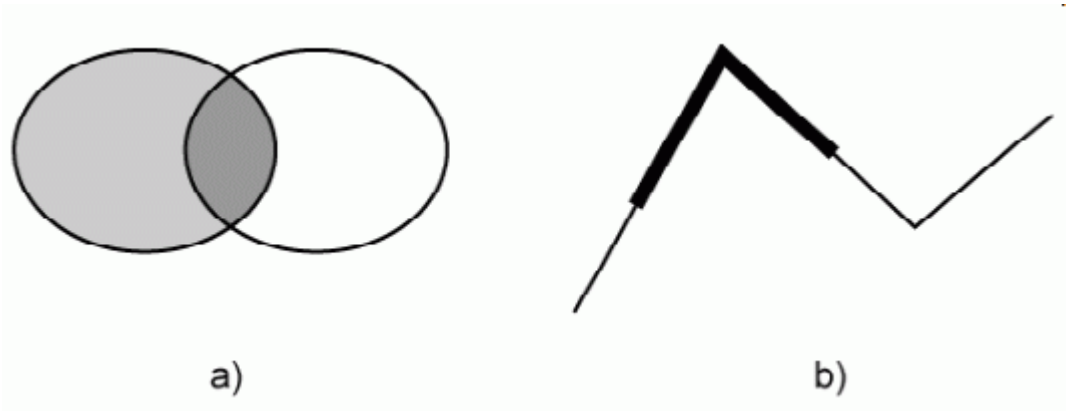


Рисунок 23: Примеры отношения Overlaps
Polygon/LineString (a), LineString/LineString(b)

Следующие дополнительные именованные предикаты также добавлены для удобства пользователя.

Contains

```
a.Contains(b) ⇔ b.Within(a)
```

Intersects

```
a.Intersects(b) ⇔ ! a.Disjoint(b)
```

Основываясь на описанных выше операторах в классе Geometry определены следующие методы:

— **Equals** (anotherGeometry: Geometry): Integer - возвращает 1 (TRUE), если данный геометрический объект совпадает с объектом anotherGeometry.

— **Disjoint** (anotherGeometry: Geometry): Integer - возвращает 1 (TRUE), если данный геометрический объект не пересекается с объектом anotherGeometry.

— **Intersects** (anotherGeometry: Geometry): Integer - возвращает 1 (TRUE), если данный геометрический объект пересекается с объектом anotherGeometry.

— **Touches** (anotherGeometry: Geometry): Integer - возвращает 1 (TRUE), если данный геометрический объект касается объекта anotherGeometry.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

— **Crosses** (anotherGeometry: Geometry): Integer - возвращает 1 (TRUE), если данный геометрический объект пространственно скрещивается с объектом anotherGeometry.

— **Within** (anotherGeometry: Geometry): Integer - возвращает 1 (TRUE), если данный геометрический объект находится внутри объекта anotherGeometry.

— **Contains** (anotherGeometry: Geometry): Integer - возвращает 1 (TRUE), если данный геометрический объект содержит объект anotherGeometry.

— **Overlaps** (AnotherGeometry: Geometry) Integer - Возвращает 1 (TRUE), если данный геометрический объект накладывается на anotherGeometry.

— **Relate** (anotherGeometry: Geometry, intersectionPatternMatrix: String): Integer - Возвращает 1 (TRUE), если данный геометрический объект имеет заданные матрицей пространственные отношения с anotherGeometry.

6.2 Текстовые подписи

ГИС приложения используют текст, размещенный на карте. Многие приложения сохраняют информацию о размещении текста в своем формате из-за отсутствия полного и удобного стандарта. Способы хранения текстовой информации стремились сделать совместимыми, но обменные форматы оказывались сложными и значительно отличались от родных форматов приложений. Поэтому их нельзя было стандартизировать для обмена сложными данными и данные обменные форматы не получили широкого распространения. Чтобы преодолеть барьер взаимодействия приложений данная спецификация использует лучшее из накопленного опыта.

Текст подписей это размещенный в пространстве текст, который может также содержать географическую или другую информацию. Данный текст может использоваться для отображения в редакторах на простых картах. Обычно он не имеет картографического качества но может применятся как упрощенное представление подписей на картах. Главная цель стандартизации подписей - дать возможность приложениям, использующим базу данных простых геометрий или XML, читать и записывать текстовые объекты, содержащие информацию о том где и как следует отображать текст. Данный стандарт гарантирует, что приложения, размещающие текст не будут иметь проблем, сохраняя результаты своей работы и приложения, использующие данную информацию, не будут иметь проблем размещая текст.

В отличии от геометрических объектов, отображение текста сильно зависит от используемых клиентом графических библиотек и от примененных к тексту атрибутов. Область, занимаемая текстом может быть лишь частично определена при помощи геометрических объектов. Необходимыми данными для корректного отображения текста являются атрибуты шрифта и геометрический объект, определяющий положение текста. Поэтому очень важно иметь возможность сохранять эту информацию в базе данных геометрии. Так как невозможно гарантировать абсолютно идентичного отображения текста различными приложениями, каждое приложение может интерпретировать данную информацию на свое усмотрение.

Главное предназначение текстовых подписей это использование их в распечатанных и отображаемых на экране художественных и технических картах. Наиболее же вероятное использование текстовых подписей это идентификация геометрических объектов. При просмотре карт перед печатью и при редактировании карт текстовые подписи отображаются с различной детализацией. Назначение текстовых подписей при этом остается тем же - облегчить восприятие картографических объектов при просмотре, редактировании и анализе карт.

Текстовые подписи могут быть использованы также для создания аннотаций. Размещенный таким образом текст может даже не быть связан с реальным объектом, а использоваться для отображения текущей служебной информации. Например, при сборе и редактировании данных, такой текст может отображать ошибки в данных, которые нуждаются в исправлении. Например, зазоры и петли при оцифровке объектов. В этих случаях подпись отображается возле места возникновения ошибки, но при этом необязательно связана с конкретным объектом, либо группой объектов.

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Аннотации могут представлять собой текст на векторных картах, либо подписи на изображениях. Подписи на изображениях могут содержать информацию, которую нельзя получить с изображения, такую как названия улиц. В большинстве случаев приложения, работающие с аннотациями имеют определенные правила для создания подписей основанных на динамических наборах. Для данного типа хранилищ допустимы определенные упрощения при отображении подписей. Хотя данная спецификация не предназначена для такого использования однако, если это необходимо, можно не масштабировать текст подписей при масштабировании объектов карты, сохраняя фиксированную высоту шрифта (выражаемую в пунктах, где 72 пункта составляют дюйм). Следует помнить, что имеются ограничения в использовании не масштабируемого текста, в частности при пространственном индексировании.

6.2.1 Текстовые объекты

Текстовый объект содержит список независимо размещаемых элементов текста, таких как строки многострочной подписи и прямоугольника, приблизительно описывающего внешние границы каждого элемента текста. Каждый элемент содержит свои текстовые атрибуты, но они не используются независимо. Первый элемент может устанавливать атрибуты для всех последующих элементов, те, в свою очередь, изменять атрибуты когда это требуется. Такой подход описывает атрибуты текста для каждого текстового элемента.

Текстовый объект состоит из строки текста и информацию о ее размещении. Наиболее важная часть информации это геометрический объект, определяющий положение текстового элемента. Здесь он называется location geometry. Второй геометрический объект может быть использован для более точного описания положения текста, особенно в случаях, когда текст окружен множеством картографических объектов. Данный геометрический объект называется здесь leader line и является кривой линией не имеющий географического значения. Если используется пространственное индексирование, минимально ограничивающий прямоугольник есть обязательная информация. Если минимальный ограничивающий прямоугольник не задан, он может быть рассчитан в при отображении текста. Однако расчет минимального ограничивающего прямоугольника это довольно трудоемкий процесс и наиболее эффективный способ работы это предварительный его расчет и сохранение в базе данных. Другая информация, связанная с текстовыми объектами это различная информация о стилях, такая как размер текста (обычно в единицах экрана пикселях или точках), имя используемого шрифта, характеристики шрифта. Это представлено на UML диаграмме на Рисунке 24.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

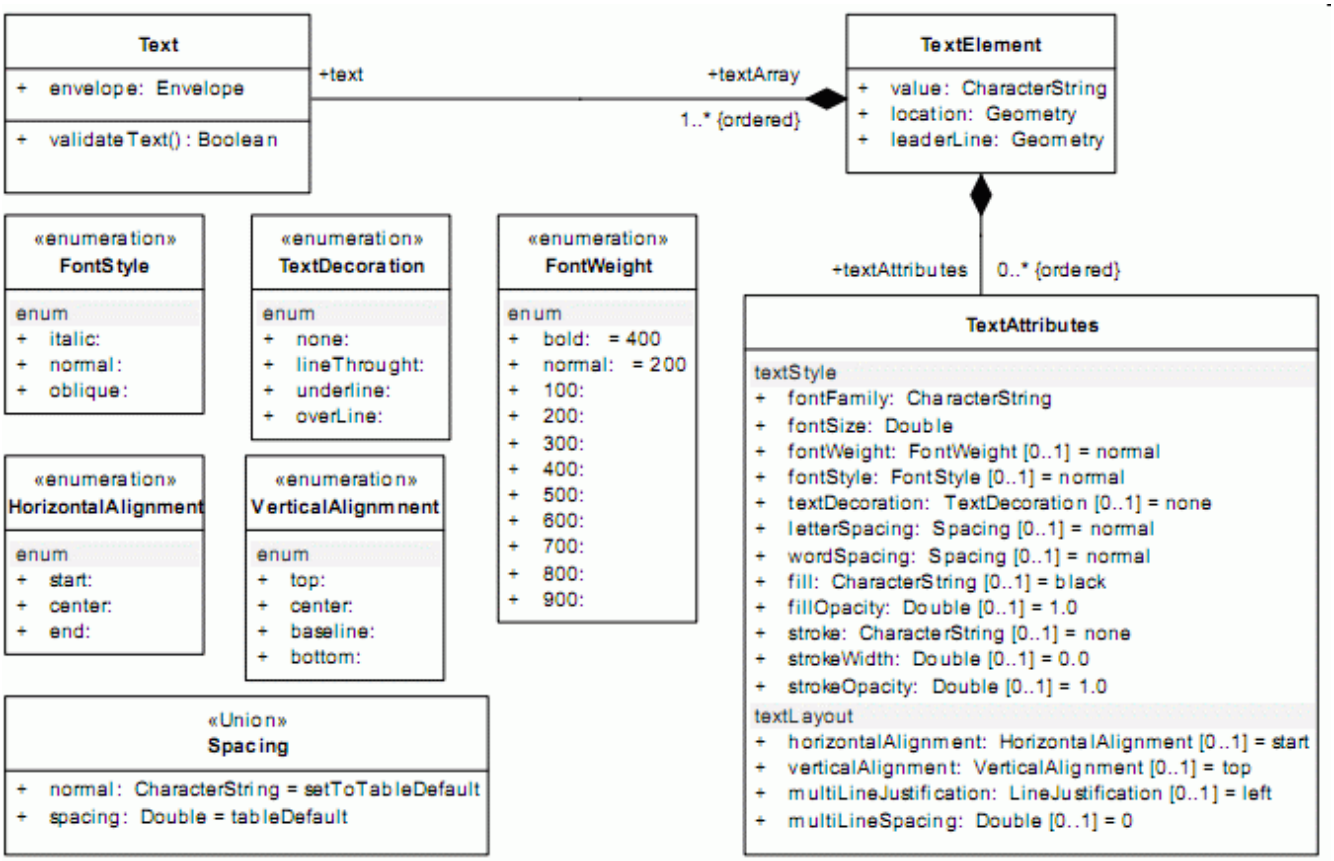


Рисунок 24: Классы текстовых объектов

Таблица 2: Поля текстового объекта

Имя поля	Тип	Требования и Умолчания
text array	Массив элементов текста ANNOTATION_TEXT_ELEMENT (данный тип описан ниже в Таблице 3)	Массив должен содержать минимум 1 элемент, иначе текст не будет отображен
envelope	GEOMETRY	Обязательное. Ограничивающий текст прямоугольник, требуемый для пространственной индексации.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Таблица 3: Поля текстового элемента

Имя поля	Тип	Требования и умолчания
value	VARCHAR2(2000)	Необязательное. Отображаемый текст. Сначала берется из значения value, если оно не пустое. Если пустое, текст берется из первого предшедствующего элемента с заполненным полем value. Если все предшествующие элементы имеют пустое значение поля value выводится значение полученное из TEXT_EXPRESSION.
location	GEOMETRY	Необязательное. Если location не задано, текст не отображается. Тип геометрического объекта может быть точка или кривая.
text attributes	XML_TYPE (строка символов в XML)	Необязательное. Однако текст не будет отображаться, если не задано минимальное количество атрибутов стиля в таблице метаданных (используется по умолчанию) либо в данном поле.
leader line	GEOMETRY (a curve type)	Необязательное. Если не задано, направляющая линия отсутствует.

6.2.2 Текстовые атрибуты

В дополнение к информации о размещении для правильного отображения текста необходима информация о его внешнем представлении. Эта информация сохраняется вместе с текстом. Текстовые атрибуты включают поля описанные в следующей таблице.

Таблица 4: Атрибуты стиля текста

Имя поля	Тип	Описание	Требования и умолчания
font-family	String	Имена такие как Arial, Helvetica, Times New Roman. Нет гарантии, что шрифт установлен в клиентской системе. Список имен должен быть отделен (;) как в SVG формате и задает упорядоченный набор имен шрифтов для использования.	Требуется непустая строка. Сервер не может проверить ее.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

font-size	Float	Размер текста, получаемый как сумма надстрочного, подстрочного и внутреннего элементов в пунктах. Обратите внимание, что данное значение используется вместе со значением масштаба карты из таблицы метаданных для которого данный размер шрифта определен. В случае если масштаб указан текст масштабируется вместе с объектами карты, иначе его размер остается постоянным. Приложение ответственно за правильный расчет текста в пикселях при его отображении.	Обязательное положительное значение.
font-weight	enumeration	Допускаются значения Normal, Bold, или 100, 200, 300, 400, 500, 600, 700, 800, 900. Normal это тоже самое, что 200. Bold то же самое, что 400.	Значение по умолчанию Normal
font-style	enumeration	Normal, Italic or Oblique.Oblique это необязательное значение в SVG формате. Определяет, что текст наклонен влево, в то время как для Italic текст наклонен вправо. Так как на самом деле стиль Oblique имеет небольшую поддержку графическими библиотеками, клиентские приложения могут использовать вместо него стиль Italic.	Значение по умолчанию Normal
text-decoration	enumeration	None, underline, line-through или over-line. При underline текст выводится на главной линии, over-line - главная линия + подъем, line-through - главная линия + подъем * 0.5. Строка рисуется цветом заполнения и окантовки, если таковой указан (см. ниже).	Значение по умолчанию None
letter-spacing	Float или “normal”	Отступы между символами. SVG позволяет использовать цифры или значение Normal.	Значение по умолчанию Normal
word-spacing	Float или “normal”	Аналогично отступам между символам, но применятся для указания отступов между словами.	Значение по умолчанию Normal

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

fill	String (тип заливки)	Поле определяет цвет заполнения символов. Цвет может быть задан следующим образом: 1. SVG имена цветов, такие как black, blue, red. См. http://www.w3.org/TR/SVG/types.html#ColorKeywords . 2. RGB значения определяемые с использованием синтаксиса, например rgb(255, 0, 255) 3. Шестнадцетиричное значение, например #FF00FF, которое обозначает такой же цвет, что и в предыдущем примере. Цвет заливки должен рассматриваться как основной цвет текста. Приложения, которые не отображают текст с окантовкой должны использовать для отображения текста с окантовкой но без заливки именно цвет заливки.	Значение по умолчанию black
fill-opacity	Float(0-1)	Коэфициент, задающий непрозрачность или полупрозрачность текста . 0 - текст полностью прозрачен, 1 - полностью непрозрачен.	Значение по умолчанию 1
stroke	String(тип штриха)	Поле задает цвет окантовки символов. Цвет окантовки может совпадать с цветом заливки. Мы предлагаем рисовать окантовку перед заливкой, это создает красивую тень вокруг символов текста.	Значение по умолчанию None.
stroke-width	Float	Значение в пунктах, определяющее ширину окантовки символов текста.	Значение по умолчанию 0, указывающее на отсутствие окантовки.
stroke-opacity	Float(0-1)	Значение, определяющее непрозрачность или полупрозрачность окантовки.	Значение по умолчанию 1.

Таблица 5: Атрибуты расположения текста

Имя поля	Тип	Описание	Требования и умолчания
horizontal alignment	Enumeration	Имеется 3 допустимsx значения: start, center и end, обозначающих какая часть текста находится возле начальной точки геометрического объекта, задающего положение текста. Например, start означает, что первый символ текста находится возле начальной точки геометрического объекта и текст будет расположен справа от этой точки.	Необязательное. Значение по умолчанию start.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

vertical alignment	Enumeration	Имеется 4 допустимых значения: top, center, baseline и bottom. Их значение такое же, что и для горизонтального выравнивания. Например, top обозначает, что верхняя часть символов текста расположена в начальной точке геометрического объекта.	Необязательное. Значение по умолчанию top.
multiline justification	Enumeration	Имеется 3 допустимsx значения: left, center и right. Эти значения определяют как линии многострочного текста буду выравниваться относительна друг друга.	Необязательное. Не требуется для однострочного текста. Значение по умолчанию left.
multiline spacing	Float	Значение в пунктах, задающее расстояние между строками текста. Измеряется от нижней части строки до верхней части нижележащей строки.	Необязательное. Не требуется для однострочного текста. Значение по умолчанию 0, при этом строки размещаются одна под другой.

6.2.3 XML для атрибутов текста

Ниже представлена XML схема для текстовых атрибутов. Используется как метаданные из таблицы метаданных текста. Схема представлена без пространства имен. Цвета представлены в SVG формате.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:complexType name="textAttributesType">
   <xs:sequence>
    <xs:element ref="textStyle"/>
    <xs:element ref="textlayout"/>
   </xs:sequence>
  </xs:complexType>
  <xs:element name="textAttributes" type="textAttributesType"/>
   <xs:element name="textStyle">
   <xs:annotation>
    <xs:documentation>Text font style attribute</xs:documentation>
   </xs:annotation>
   <xs:complexType>
    <xs:attribute name="font-family" type="xs:string" use="required"/>
    <xs:attribute name="font-size" type="xs:float" use="required"/>
    <xs:attribute name="font-weight" type="fontWeight" use="optional"
default="Normal"/>
    <xs:attribute name="font-style" type="fontStyle" use="optional"
default="Normal"/>
    <xs:attribute name="text-decoration" type="textDecoration" use="optional"
default="None"/>
    <xs:attribute name="letter-spacing" use="optional" default="Normal"/>
```


Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
    <xs:attribute name="word-spacing" type="spacing" use="optional"
default="Normal"/>
    <xs:attribute name="fill" type="colorType" use="optional" default="black"/>
    <xs:attribute name="fill-opacity" type="opacity" use="optional" default="1.0"/>
    <xs:attribute name="stroke" type="colorType" use="optional" default="black"/>
    <xs:attribute name="stroke-width" type="xs:float" use="optional"
default="1.0"/>
    <xs:attribute name="stroke-opacity" type="opacity" use="optional"
default="1.0"/>
   </xs:complexType>
  </xs:element>
  <xs:element name="textlayout">
   <xs:annotation>
    <xs:documentation>Text alignment and justification </xs:documentation>
   </xs:annotation>
   <xs:complexType>
    <xs:attribute name="horizontalAlignment" use="optional" default="start">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:enumeration value="start"/>
       <xs:enumeration value="center"/>
       <xs:enumeration value="end"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="verticalAlignment" use="optional" default="top">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:enumeration value="top"/>
       <xs:enumeration value="center"/>
       <xs:enumeration value="baseline"/>
       <xs:enumeration value="bottom"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="multilineJustification" use="optional" default="left">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:enumeration value="left"/>
       <xs:enumeration value="center"/>
       <xs:enumeration value="right"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="multilineSpacing" type="xs:float" use="optional"
default="0.0"/>
   </xs:complexType>
  </xs:element>
  <xs:simpleType name="fontWeight">
   <xs:restriction base="xs:string">
    <xs:enumeration value="Normal"/>
    <xs:enumeration value="Bold"/>
    <xs:enumeration value="100"/>
    <xs:enumeration value="200"/>
    <xs:enumeration value="300"/>
    <xs:enumeration value="400"/>
    <xs:enumeration value="500"/>
    <xs:enumeration value="600"/>
    <xs:enumeration value="700"/>
```

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
    <xs:enumeration value="800"/>
    <xs:enumeration value="900"/>
   </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="fontStyle">
   <xs:restriction base="xs:string">
    <xs:enumeration value="Normal"/>
    <xs:enumeration value="Italics"/>
    <xs:enumeration value="Oblique"/>
   </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="textDecoration">
   <xs:restriction base="xs:string">
    <xs:enumeration value="None"/>
    <xs:enumeration value="Underline"/>
    <xs:enumeration value="LineThrough"/>
    <xs:enumeration value="Overline"/>
   </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="spacing">
   <xs:union>
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:enumeration value="Normal"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
     <xs:restriction base="xs:float"/>
    </xs:simpleType>
   </xs:union>
  </xs:simpleType>
  <xs:simpleType name="colorType">
   <xs:union>
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:pattern value="(rgb\(N,N,N\))"/>
     </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:enumeration value="none"/>
      <xs:enumeration value="aliceblue"/>
      <xs:enumeration value="antiquewhite"/>
      <xs:enumeration value="aqua"/>
      <xs:enumeration value="aquamarine"/>
      <xs:enumeration value="azure"/>
      <xs:enumeration value="beige"/>
      <xs:enumeration value="bisque"/>
      <xs:enumeration value="black"/>
      <xs:enumeration value="blanchedalmond"/>
      <xs:enumeration value="blue"/>
      <xs:enumeration value="blueviolet"/>
      <xs:enumeration value="brown"/>
      <xs:enumeration value="burlywood"/>
      <xs:enumeration value="cadetblue"/>
      <xs:enumeration value="chartreuse"/>
      <xs:enumeration value="chocolate"/>
      <xs:enumeration value="coral"/>
      <xs:enumeration value="cornflowerblue"/>
```


Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
<xs:enumeration value="cornsilk"/>
<xs:enumeration value="crimson"/>
<xs:enumeration value="cyan"/>
<xs:enumeration value="darkblue"/>
<xs:enumeration value="darkcyan"/>
<xs:enumeration value="darkgoldenrod"/>
<xs:enumeration value="darkgray"/>
<xs:enumeration value="darkgreen"/>
<xs:enumeration value="darkgrey"/>
<xs:enumeration value="darkkhaki"/>
<xs:enumeration value="darkmagenta"/>
<xs:enumeration value="darkolivegreen"/>
<xs:enumeration value="darkorange"/>
<xs:enumeration value="darkorchid"/>
<xs:enumeration value="darkred"/>
<xs:enumeration value="darksalmon"/>
<xs:enumeration value="darkseagreen"/>
<xs:enumeration value="darkslateblue"/>
<xs:enumeration value="darkslategray"/>
<xs:enumeration value="darkslategrey"/>
<xs:enumeration value="darkturquoise"/>
<xs:enumeration value="darkviolet"/>
<xs:enumeration value="deeppink"/>
<xs:enumeration value="deepskyblue"/>
<xs:enumeration value="dimgray"/>
<xs:enumeration value="dimgrey"/>
<xs:enumeration value="dodgerblue"/>
<xs:enumeration value="firebrick"/>
<xs:enumeration value="floralwhite"/>
<xs:enumeration value="forestgreen"/>
<xs:enumeration value="fuchsia"/>
<xs:enumeration value="gainsboro"/>
<xs:enumeration value="ghostwhite"/>
<xs:enumeration value="gold"/>
<xs:enumeration value="goldenrod"/>
<xs:enumeration value="gray"/>
<xs:enumeration value="grey"/>
<xs:enumeration value="green"/>
<xs:enumeration value="greenyellow"/>
<xs:enumeration value="honeydew"/>
<xs:enumeration value="hotpink"/>
<xs:enumeration value="indianred"/>
<xs:enumeration value="indigo"/>
<xs:enumeration value="ivory"/>
<xs:enumeration value="khaki"/>
<xs:enumeration value="lavender"/>
<xs:enumeration value="lavenderblush"/>
<xs:enumeration value="lawngreen"/>
<xs:enumeration value="lemonchiffon"/>
<xs:enumeration value="lightblue"/>
<xs:enumeration value="lightcoral"/>
<xs:enumeration value="lightcyan"/>
<xs:enumeration value="lightgoldenrodyellow"/>
<xs:enumeration value="lightgray"/>
<xs:enumeration value="lightgreen"/>
<xs:enumeration value="lightgrey"/>
<xs:enumeration value="lightpink"/>
<xs:enumeration value="lightsalmon"/>
<xs:enumeration value="lightseagreen"/>
```

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
<xs:enumeration value="lightskyblue"/>
<xs:enumeration value="lightslategray"/>
<xs:enumeration value="lightslategrey"/>
<xs:enumeration value="lightsteelblue"/>
<xs:enumeration value="lightyellow"/>
<xs:enumeration value="lime"/>
<xs:enumeration value="limegreen"/>
<xs:enumeration value="linen"/>
<xs:enumeration value="magenta"/>
<xs:enumeration value="maroon"/>
<xs:enumeration value="mediumaquamarine"/>
<xs:enumeration value="mediumblue"/>
<xs:enumeration value="mediumorchid"/>
<xs:enumeration value="mediumpurple"/>
<xs:enumeration value="mediumseagreen"/>
<xs:enumeration value="mediumslateblue"/>
<xs:enumeration value="mediumspringgreen"/>
<xs:enumeration value="mediumturquoise"/>
<xs:enumeration value="mediumvioletred"/>
<xs:enumeration value="midnightblue"/>
<xs:enumeration value="mintcream"/>
<xs:enumeration value="mistyrose"/>
<xs:enumeration value="moccasin"/>
<xs:enumeration value="navajowhite"/>
<xs:enumeration value="navy"/>
<xs:enumeration value="oldlace"/>
<xs:enumeration value="olive"/>
<xs:enumeration value="olivedrab"/>
<xs:enumeration value="orange"/>
<xs:enumeration value="orangered"/>
<xs:enumeration value="orchid"/>
<xs:enumeration value="palegoldenrod"/>
<xs:enumeration value="palegreen"/>
<xs:enumeration value="paleturquoise"/>
<xs:enumeration value="palevioletred"/>
<xs:enumeration value="papayawhip"/>
<xs:enumeration value="peachpuff"/>
<xs:enumeration value="peru"/>
<xs:enumeration value="pink"/>
<xs:enumeration value="plum"/>
<xs:enumeration value="powderblue"/>
<xs:enumeration value="purple"/>
<xs:enumeration value="red"/>
<xs:enumeration value="rosybrown"/>
<xs:enumeration value="royalblue"/>
<xs:enumeration value="saddlebrown"/>
<xs:enumeration value="salmon"/>
<xs:enumeration value="sandybrown"/>
<xs:enumeration value="seagreen"/>
<xs:enumeration value="seashell"/>
<xs:enumeration value="sienna"/>
<xs:enumeration value="silver"/>
<xs:enumeration value="skyblue"/>
<xs:enumeration value="slateblue"/>
<xs:enumeration value="slategray"/>
<xs:enumeration value="slategrey"/>
<xs:enumeration value="snow"/>
<xs:enumeration value="springgreen"/>
<xs:enumeration value="steelblue"/>
```


Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
      <xs:enumeration value="tan"/>
      <xs:enumeration value="teal"/>
      <xs:enumeration value="thistle"/>
      <xs:enumeration value="tomato"/>
      <xs:enumeration value="turquoise"/>
      <xs:enumeration value="violet"/>
      <xs:enumeration value="wheat"/>
      <xs:enumeration value="white"/>
      <xs:enumeration value="whitesmoke"/>
      <xs:enumeration value="yellow"/>
      <xs:enumeration value="yellowgreen"/>
     </xs:restriction>
    </xs:simpleType>
   </xs:union>
  </xs:simpleType>
  <xs:simpleType name="opacity">
   <xs:restriction base="xs:float">
    <xs:minInclusive value="0.0"/>
    <xs:maxInclusive value="1.0"/>
   </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

7 WKT-представление объектов Geometry

7.1 Обзор

Каждый Тип объекта Geometry имеет свое представление в текстовом виде (далее – представление WKT), которое может быть использовано как для создания новых экземпляров объектов данного типа, так и для преобразования существующих экземпляров объектов в текстовый вид для буквенно-цифрового отображения.

7.2 Описание

7.2.1 Представление BNF[4]

Представление WKT Геометрии определено ниже с использованием BNF.

- Обозначение «{}» указывает на необязательный элемент[5] внутри скобок. Скобки в выходном списке элементов не отображаются.
- Обозначение «()» группирует последовательность элементов в один элемент. Также не отображается в выходном списке.
- Обозначение «*» после элемента указывает на возможное использование нескольких экземпляров данного элемента[6].
- Строка символов без специальных обозначений определяет строку как элемент.
- Обозначение «|» определяет выбор между двумя элементами[7] и не отображается в выходном списке.
- Обозначение «< >» содержит производный тип либо основной тип, определенный где-либо в списке.
- Обозначение «:=» предполагает замену выражения слева от обозначения на выражение справа от обозначение. Операция прекращается, когда выражению слева приравнивается неопределенное ранее значение.

Текстовое представление геометрии должно базироваться на описанных принципах. Строка WKT нечувствительна к регистру. Поэтому, когда важна читабельность, может быть использован верхний регистр (т.е. написание служебных слов заглавными буквами, см. примеры в данной спецификации).

Примечание: Все производные строки (productions) разделяются по координатному типу. Это значит, что любые два субэлемента любого элемента будут иметь такой же координатный тип, что и родительский элемент.

Правила, описанные в этом и предыдущем разделах, позволяют компактно и читабельно описывать текстовые представления геометрии объектов. Представление геометрии объектов, состоящих из однородных компонентов, не содержит тэгов каждого дочернего компонента. Нижеследующий набор строк описывает представление чисел двойной точности в текстовом виде.

```
<x> ::=                          <signed numeric literal>

<y> ::=                          <signed numeric literal>

<z> ::=                          <signed numeric literal>

<m> ::=                          <signed numeric literal>

<quoted name> ::=                <double quote> <name> <double quote>
```

[4] BNF (Backus-Naur Form) – это формальный математический способ описания языка. По-сути это язык описания языка. Использовался в языке программирования Algol 60.
[5] Token – элемент языка, символ, слово, обозначение.
[6] Например, (<digit>)* - это и 4 и 41928.
[7] Например, <plus sign> | <minus sign> - это либо знак плюс, либо знак минус, но не два знака одновременно.

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
<name> ::=                          <letters>

<letters> ::=                       (<letter>)*

<quoted name> ::=                   <double quote> <name> <double quote>

<name> ::=                          <letters>

<letters> ::=                       (<letter>)*

<letter> ::=                        <simple Latin letter>|<digit>|<special>

<simple Latin letter> ::=           <simple Latin upper case letter>
                                   |<simple Latin lower case letter>

<signed numeric literal> ::=        {<sign>}<unsigned numeric literal>

<unsigned numeric literal> ::=      <exact numeric literal>
                                   |<approximate numeric literal>

<approximate numeric literal> ::=   <mantissa>E<exponent>

<mantissa> ::=                      <exact numeric literal>

<exponent> ::=                      <signed integer>

<exact numeric literal> ::=         <unsigned integer>
                                   {<decimal point>{<unsigned integer>}}
                                   |<decimal point><unsigned integer>

<signed integer> ::=                {<sign>}<unsigned integer>

<unsigned integer> ::=              (<digit>)*

<left delimiter> ::=                <left paren>|<left bracket>
                                   // должен быть выровнен по правому разделителю

<right delimiter> ::=               <right paren>|<right bracket>
                                   // должен быть выровнен по левому разделителю

<special> ::=                       <right paren>|<left paren>|<minus sign>
                                   |<underscore>|<period>|<quote>|<space>

<sign> ::=                          <plus sign> | <minus sign>

<decimal point> ::=                 <period> | <comma>

<empty set> ::=                     EMPTY

<minus sign> ::=                    -

<left paren> ::=                    (

<right paren> ::=                   )

<left bracket> ::=                  [

<right bracket> ::=                 ]
```

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
<period> ::=                        .

<plus sign> ::=                     +

<double quote> ::=                  "

<quote> ::=                         '

<comma> ::=                         ,

<underscore> ::=                    _

<digit> ::=                         0|1|2|3|4|5|6|7|8|9

<simple Latin lower case            a|b|c|d|e|f|g|h|i|j|k|l|m
      letter> ::=                   |n|o|p|q|r|s|t|u|v|w|x|y|z

<simple Latin upper case            A|B|C|D|E|F|G|H|I|J|K|L|M
      letter> ::=                   |N|O|P|Q|R|S|T|U|V|W|X|Y|Z

<space>=                            " "
                                    // Юникод "U+0020" (пробел)
```

7.2.2 BNF-описание для Двумерной WKT Geometry

Данное BNF-описагие определяет двумерную геометрию в (x, y) координатной плоскости. За исключением добавленных многогранных поверхностей, эти структуры не отличаются от опубликованных в ранних версиях данной спецификации.

```
<point> ::=                         <x> <y>

<geometry tagged text> ::=          <point tagged text>
                                    | <linestring tagged text>
                                    | <polygon tagged text>
                                    | <triangle tagged text>
                                    | <polyhedralsurface tagged text>
                                    | <tin tagged text>
                                    | <multipoint tagged text>
                                    | <multilinestring tagged text>
                                    | <multipolygon tagged text>
                                    | <geometrycollection tagged text>

<point tagged text> ::=             point <point text>

<linestring tagged text> ::=        linestring <linestring text>

<polygon tagged text> ::=           polygon <polygon text>

<polyhedralsurface tagged           polyhedralsurface
      text> ::=                     <polyhedralsurface text>

<triangle tagged text> ::=          triangle <polygon text>

<tin tagged text> ::=               tin <polyhedralsurface text>

<multipoint tagged text> ::=        multipoint <multipoint text>
```

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
<multilinestring tagged text> ::=   multilinestring
                                    <multilinestring text>

<multipolygon tagged text> ::=      multipolygon <multipolygon text>

<geometrycollection tagged          geometrycollection
      text> ::=                     <geometrycollection text>

<point text> ::=                    <empty set>|<left paren> <point> <right paren>

<linestring text> ::=               <empty set> | <left paren>
                                    <point>
                                    {<comma> <point>}*
                                    <right paren>

<polygon text> ::=                  <empty set> | <left paren>
                                    <linestring text>
                                    {<comma> <linestring text>}*
                                    <right paren>

<polyhedralsurface text> ::=        <empty set> | <left paren>
                                    <polygon text>
                                    {<comma> <polygon text>}*
                                    <right paren>

<multipoint text> ::=               <empty set> | <left paren>
                                    <point text>
                                    {<comma> <point text>}*
                                    <right paren>

<multilinestring text> ::=          <empty set> | <left paren>
                                    <linestring text>
                                    {<comma> <linestring text>}*
                                    <right paren>

<multipolygon text> ::=             <empty set> | <left paren>
                                    <polygon text>
                                    {<comma> <polygon text>}*
                                    <right paren>

<geometrycollection text> ::=       <empty set> | <left paren>
                                    <geometry tagged text>
                                    {<comma> <geometry tagged text>}*
                                    <right paren>
```

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

7.2.3 BNF-описание для Трехмерной WKT Geometry

Этот BNF-описание определяет геометрию в трехмерных (x, y, z) координатах.

```
<point z> ::=                          <x> <y> <z>

<geometry z tagged text> ::=           <point z tagged text>
                                      |<linestring z tagged text>
                                      |<polygon z tagged text>
                                      |<polyhedralsurface z tagged text>
                                      |<triangle tagged text>
                                      |<tin tagged text>
                                      |<multipoint z tagged text>
                                      |<multilinestring z tagged text>
                                      |<multipolygon z tagged text>
                                      |<geometrycollection z tagged text>

<point z tagged text> ::=              point z <point z text>

<linestring z tagged text> ::=         linestring z <linestring z text>

<polygon z tagged text> ::=            polygon z <polygon z text>

<polyhedralsurface z tagged            polyhedralsurface z
      text> ::=                        <polyhedralsurface z text>

<triangle z tagged text> ::=           triangle z <polygon z text>

<tin z tagged text>                    tin z <polyhedralsurface z text>

<multipoint z tagged text> ::=         multipoint z <multipoint z text>

<multilinestring z tagged              multilinestring z <multilinestring z text>
text> ::=

<multipolygon z tagged text> ::=       multipolygon z <multipolygon z text>

<geometrycollection z tagged           geometrycollection z
      text> ::=                        <geometrycollection z text>

<point z text> ::=                     <empty set> | <left paren> <point z>
                                      <right paren>

<linestring z text> ::=                <empty set> | <left paren> <point z>
                                      {<comma> <point z>}*
                                      <right paren>

<polygon z text> ::=                   <empty set> | <left paren>
                                      <linestring z text>
                                      {<comma> <linestring z text>}*
                                      <right paren>

<polyhedralsurface z text> ::=         <empty set>|<left paren>
                                      <polygon z text>
                                      {<comma> <polygon z text>}*
                                      <right paren>

<multipoint z text> ::=                <empty set> | <left paren>
```


Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
                                    <point z text>
                                    {<comma> <point z text>}*
                                    <right paren>

<multilinestring z text> ::=        <empty set> | <left paren>
                                    <linestring z text>
                                    {<comma> <linestring z text>}*
                                    <right paren>

<multipolygon z text> ::=           <empty set> | <left paren>
                                    <polygon z text>
                                    {<comma> <polygon z text>}*
                                    <right paren>

<geometrycollection z text> ::=     <empty set> | <left paren>
                                    <geometry tagged z text>
                                    {<comma> <geometry tagged z text>}*
                                    <right paren>
```

7.2.4 BNF-описание для Двухмерной WKT Geometry с измереным значением в точке

Данное BNF-описание определяет двумерную геометрию в (x, y) координатной плоскости. Дополнительно, каждая координата имеет ординату m, которая представляет собой некую величину, соответствующую данной точке.

```
<point m> ::=                       <x> <y> <m>

<geometry m tagged text> ::=        <point m tagged text>
                                   |<linestring m tagged text>
                                   |<polygon m tagged text>
                                   |<polyhedralsurface m tagged text>
                                   |<triangle tagged m text>
                                   |<tin tagged m text>
                                   |<multipoint m tagged text>
                                   |<multilinestring m tagged text>
                                   |<multipolygon m tagged text>
                                   |<geometrycollection m tagged text>

<point m tagged text> ::=           point m <point m text>

<linestring m tagged text> ::=      linestring m <linestring m text>

<polygon m tagged text> ::=         polygon m <polygon m text>

<polyhedralsurface m tagged         polyhedralsurface m
      text> ::=                     <polyhedralsurface m text>

<triangle m tagged text> ::=        triangle m <polygon m text>

<tin m tagged text>                 tin m <polyhedralsurface m text>

<multipoint m tagged text> ::=      multipoint m <multipoint m text>

<multilinestring m tagged           multilinestring m <multilinestring m text>
      text> ::=

<multipolygon m tagged text> ::=    multipolygon m <multipolygon m text>

<geometrycollection m tagged        geometrycollection m
```

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
      text> ::=                    <geometrycollection m text>

<point m text> ::=                 <empty set> | <left paren>
                                   <point m>
                                   <right paren>

<linestring m text> ::=            <empty set> | <left paren>
                                   <point m>
                                   {{<comma> <point m>}+
                                   <right paren>

<polygon m text> ::=               <empty set> | <left paren>
                                   <linestring m text>
                                   {<comma> <linestring m text>}*
                                   <right paren>

<polyhedralsurface m text> ::=     <empty set> | <left paren>
                                   <polygon m text>
                                   {<comma> <polygon m text>}*
                                   <right paren>

<multipoint m text> ::=            <empty set> | <left paren> <point m text>
                                   {<comma> <point m text>}*
                                   <right paren>

<multilinestring m text> ::=       <empty set> | <left paren>
                                   <linestring m text>
                                   {<comma> <linestring m text>}*
                                   <right paren>

<multipolygon m text> ::=          <empty set> | <left paren>
                                   <polygon m text>
                                   {<comma> <polygon m text>}*
                                   <right paren>

<geometrycollection m text> ::=    <empty set> | <left paren>
                                   <geometry tagged m text>
                                   {<comma> <geometry tagged m text>}*
                                   <right paren>
```

7.2.5 BNF-описание для Трехмерной WKT Geometry с m-значениями

Данное BNF-описание определяет трехмерную геометрию в (x, y, z) координатной плоскости. Дополнительно, каждая координата имеет ординату m, которая представляет собой некую величину, соответствующую данной точке.

```
<point zm> ::=                     <x> <y> <z> <m>

<geometry zm tagged text> ::=      <point zm tagged text>
                                   |<linestring zm tagged text>
                                   |<polygon zm tagged text>
                                   |<polyhedralsurface zm tagged text>
                                   |<triangle zm tagged text>
                                   |<tin zm tagged text>
                                   |<multipoint zm tagged text>
                                   |<multilinestring zm tagged text>
                                   |<multipolygon zm tagged text>
                                   |<geometrycollection zm tagged text>
```

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
<point zm tagged text> ::=          point zm <point zm text>

<linestring zm tagged text> ::=     linestring zm <linestring zm text>

<polygon zm tagged text> ::=        polygon zm <polygon zm text>

<polyhedralsurface zm tagged        polyhedralsurface zm
      text> ::=                     <polyhedralsurface zm text>

<triangle zm tagged text> ::=       triangle zm <polygon zm text>

<tin zm tagged text>                tin zm <polyhedralsurface zm text>

<multipoint zm tagged text> ::=     multipoint zm <multipoint zm text>

<multipoint zm tagged text> ::=     multipoint zm <multipoint zm text>

<multilinestring zm tagged          multilinestring zm
      text> ::=                     <multilinestring zm text>

<multipolygon zm tagged text> ::=   multipolygon zm <MultiPolygon zm text>

<geometrycollection zm tagged       geometrycollection zm
      text> ::=                     <geometrycollection zm text>

<point zm text> ::=                 <empty set> | <left paren> <point zm>
                                    <right paren>

<linestring zm text> ::=            <empty set> | <left paren>
                                    <point z>
                                    {<comma> <point z>}*
                                    <right paren>

<polygon zm text> ::=               <empty set> | <left paren>
                                    <linestring zm text>
                                    {<comma> <linestring zm text>}*
                                    <right paren>

<polyhedralsurface zm text> ::=     <empty set> | <left paren> {
                                    <polygon zm text
                                    {<comma> <polygon zm text>}*)
                                    <right paren>

<multipoint zm text> ::=            <empty set> | <left paren>
                                    <point zm text>
                                    {<comma> <point zm text>}*
                                    <right paren>

<multilinestring zm text> ::=       <empty set> | <left paren>
                                    <linestring zm text>
                                    {<comma> <linestring zm text>}*
                                    <right paren>

<multipolygon zm text> ::=          <empty set> | <left paren>
                                    <polygon zm text>
                                    {<comma> <polygon zm text>}*
                                    <right paren>

<geometrycollection zm text> ::=    <empty set> | <left paren>
```

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
<geometry tagged zm text>
{<comma> <geometry tagged zm text>}*
<right paren>
```

7.2.6 Примеры

Примеры текстового представления Geometry приведены в Таблице 2. Координаты представляют собой целочисленные значения; они также могут принимать любые значения с двойной точностью.

Примечание: Примеры POINTZ, POINTM и POINTZM приведены внизу таблицы 6. Этот же способ различения 2D/3D точек и 2D/3D точек с координатой M может быть использован для типов LINESTRING, POLYGON, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON и GEOMETRYCOLLECTION.

Таблица 6: Примеры WKT-представления Геометрии

Тип Geometry	Текстовое представление	Комментарии
Point	Point (10 10)	Точка
LineString	LineString (10 10, 20 20, 30 40)	Ломаная линия из 3 точек
Polygon	Polygon ((10 10, 10 20, 20 20, 20 15, 10 10))	Полигон с 1 внешним кольцом и 0 внутренним кольцом
MiltiPoint	MultiPoint ((10 10), (20 20))	Мультиточка из 2 точек
MultiLineString	MultiLineString ((10 10, 20 20), (15 15, 30 15))	Ломаная мультилиния из 2 ломаных линий
MultiPolygon	MultiPolygon (((10 10, 10 20, 20 20, 20 15, 10 10)), ((60 60, 70 70, 80 60, 60 60)))	Мультиполигон, состоящий из 2 полигонов
GeomCollection	GeometryCollection (POINT (10 10), POINT (30, 30), LINESTRING (15 15, 20 20))	Коллекция, содержащая 2 точечных объекта и 1 ломаную линию
PolyhedralSurface	PolyhedralSurface Z (((0 0 0, 0 0 1, 0 1 1, 0 1 0, 0 0 0)), ((0 0 0, 0 1 0, 1 1 0, 1 0 0, 0 0 0)), ((0 0 0, 1 0 0, 1 0 1, 0 0 1, 0 0 0)), ((1 1 0, 1 1 1, 1 0 1, 1 0 0, 1 1 0)), ((0 1 0, 0 1 1, 1 1 1, 0 1 1, 0 0 1)), ((0 0 1, 1 0 1, 1 1 1, 0 1 1, 0 0 1)))	Многогранный куб, начальный угол с координатами (0, 0, 0), противоположный – (1, 1, 1)
Tin	Tin Z (((0 0 0, 0 0 1, 0 1 0, 0 0 0)), ((0 0 0, 0 1 0, 1 0 0, 0 0 0)), ((0 0 0, 1 0 0, 0 0 1, 0 0 0)), ((1 0 0, 0 1 0, 0 0 1, 1 0 0)))	Тетраэдр (4 треугольные грани)
Point	Point Z (10 10 5)	3D точка
Point	Point ZM (10 10 5 40)	3D точка со значением M 40
Point	Point M (10 10 40)	2D точка со значением M 40

8 WKB-представление объектов Geometry

8.1 Обзор

Бинарное представление объекта Geometry (WKBGeometry) предусматривает доступное представление геометрического объекта как непрерывный поток байтов. Это позволяет выполнять обмен геометрическими объектами между SQL/CLT клиентом и SQL-реализацией в бинарной форме.

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

8.2 Описание

8.2.1 Представление

WKB-представление Геометрии получается преобразованием геометрического объекта как последовательности числовых типов, отрисованных из набора {Unsigned Integer, Double} и последующим преобразовании каждого числового типа в последовательность байтов используя одно из двух правильно определенных, стандартных, бинарных представлений числовых типов (NDR, XDR). Специальная бинарная кодировка (NDR или XDR), используемая для представления геометрии, описана однобайтным тэгом, который предшествует преобразованию байтов. Разница между двумя кодировками геометрии заключается в том, что последовательность байтов кодируется обратным порядком байтов (Big Endian) в XDR и прямым порядком байтов (Little Endian) в NDR.

8.2.2 Определение числового типа

`Unsigned Integer` – 32-битный (4-байтный) тип данных, который кодируется как неотрицательное целое число в интервале [0, 4, 294, 967, 295].

`Double` – 64-битный (8-байтный) тип данных двойной точности, который кодируется числами двойной точности с использованием IEEE формата двойной точности 754[18].

Эти определения являются общими для XDR и NDR.

8.2.3 Общий список кодов типов геометрии

В этой и других главах спецификации типы геометрии идентифицированы целочисленными кодами (integer codes). Для одновременного хранения таких кодов и резервирования секций для дальнейшего использования был описан список всех типов геометрических объектов данной спецификации или запланированных реализовать в дальнейшем. Ниже приведена таблица, в которой затененными представлены коды, реализация которых предусмотрена в будущем и которые не используют описанные в спецификации типы.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Таблица 7: Коды типов объектов

Type	Code
Geometry	0
Point	1
LineString	2
Polygon	3
MultiPoint	4
MultiLineString	5
MultiPolygon	6
GeometryCollection	7
CircularString	8
CompoundCurve	9
CurvePolygon	10
MultiCurve	11
MultiSurface	12
Curve	13
Surface	14
PolyhedralSurface	15
TIN	16

Type	Code
Geometry Z	1000
Point Z	1001
LineString Z	1002
Polygon Z	1003
MultiPoint Z	1004
MultiLineString Z	1005
MultiPolygon Z	1006
GeometryCollection Z	1007
CircularString Z	1008
CompoundCurve Z	1009
CurvePolygon Z	1010
MultiCurve Z	1011
MultiSurface Z	1012
Curve Z	1013
Surface Z	1014
PolyhedralSurface Z	1015
TIN Z	1016

Type	Code
Geometry M	2000
Point M	2001
LineString M	2002
Polygon M	2003
MultiPoint M	2004
MultiLineString M	2005
MultiPolygon M	2006
GeometryCollection M	2007
CircularString M	2008
CompoundCurve M	2009
CurvePolygon M	2010
MultiCurve M	2011
MultiSurface M	2012
Curve M	2013
Surface M	2014
PolyhedralSurface M	2015
TIN M	2016

Type	Code
Geometry ZM	3000
Point ZM	3001
LineString ZM	3002
Polygon ZM	3003
MultiPoint ZM	3004
MultiLineString ZM	3005
MultiPolygon ZM	3006
GeometryCollection ZM	3007
CircularString ZM	3008
CompoundCurve ZM	3009
CurvePolygon ZM	3010
MultiCurve ZM	3011
MultiSurface ZM	3012
Curve ZM	3013
Surface ZM	3014
PolyhedralSurface ZM	3015
TIN ZM	3016

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

8.2.4 XDR (Big Endian) кодирование числовых типов

XDR-представление `Unsigned Integer` заключается в обратном порядке байтов.

XDR-представление `Double` заключается в обратном порядке байтов.

8.2.5 NDR (Little Endian) кодирование числовых типов

NDR-представление `Unsigned Integer` заключается в прямом порядке байтов.

NDR-представление `Double` заключается в прямом порядке байтов.

8.2.6 Преобразования между типами NDR и XDR

Преобразования между типами NDR и XDR для `Unsigned Integer` и `Double` является простой операцией обращения порядка байтов в каждом `Unsigned Integer` или `Double` в представлении.

8.2.7 Взаимосвязь с протоколами передачи данных COM и CORBA

Описанное выше XDR-представление для `Unsigned Integer` и `Double` является так же стандартным представлением для `Unsigned Integer` и `Double` в CORBA Standard Stream Format для Externalized Object Data, которое описано как часть CORBA Externalization Service Specification [15].

NDR-представление `Unsigned Integer` и `Double`, описанное выше, так же является стандартом представления для `Unsigned Integer` и `Double` чисел в протоколах DCOM, основанных на DCE RPC и NDR [16].

8.2.8 Описание представления WKBGeometry

В этой главе описывается бинарное представление Geometry (WKBGeometry). Основным строительным элементом является представление Point, состоящей из чисел Doubles, которое зависит от системы координат, принятой для отображения геометрии. Представление других геометрических объектов базируется на использовании уже описанных представлений.

```
// Basic Type definitions
// byte : 1 byte
// uint32 : 32 bit unsigned integer (4 bytes)
// double : double precision number (8 bytes)
// Building Blocks : Coordinate, LinearRing
Point {
   double x;
   double y}
PointZ{
   double x;
   double y;
   double z}
PointM {
   double x;
   double y;
   double m}
PointZM {
   double x;
   double y;
   double z;
   double m}
LinearRing {
```

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
   uint32 numPoints;
   Point points[numPoints]}
LinearRingZ {
   uint32 numPoints;
   PointZ points[numPoints]}
LinearRingM {
   uint32 numPoints;
   PointM points[numPoints]}
LinearRingZM {
   uint32 numPoints;

   PointZM points[numPoints]}
enum WKBByteOrder {
   wkbXDR = 0, // Big Endian
   wkbNDR = 1, // Little Endian
}
enum WKBGeometryType {
               wkbPoint                  = 1,
               wkbLineString             = 2,
               wkbPolygon                = 3,
               wkbTriangle               = 17,
               wkbMultiPoint             = 4,
               wkbMultiLineString        = 5,
               wkbMultiPolygon           = 6,
               wkbGeometryCollection     = 7,
               wkbPolyhedralSurface = 15,
               wkbTIN                    = 16

               wkbPointZ                      = 1001,
               wkbLineStringZ            = 1002,
               wkbPolygonZ               = 1003,
               wkbTrianglez              = 1017,
               wkbMultiPointZ            = 1004,
               wkbMultiLineStringZ       = 1005,
               wkbMultiPolygonZ          = 1006,
               wkbGeometryCollectionZ    = 1007,
               wkbPolyhedralSurfaceZ     = 1015,
               wkbTINZ                   = 1016

               wkbPointM                      = 2001,
               wkbLineStringM            = 2002,
               wkbPolygonM               = 2003,
               wkbTriangleM              = 2017
               wkbMultiPointM            = 2004,
               wkbMultiLineStringM       = 2005,
               wkbMultiPolygonM          = 2006,
               wkbGeometryCollectionM    = 2007,
               wkbPolyhedralSurfaceM     = 2015,
               wkbTINM                   = 2016,

               wkbPointZM                = 3001,
               wkbLineStringZM           = 3002,
               wkbPolygonZM              = 3003,
               wkbTriangleZM             = 3017,
               wkbMultiPointZM           = 3004,
               wkbMultiLineStringZM = 3005,
               wkbMultiPolygonZM         = 3006,
               wkbGeometryCollectionZM   = 3007,
               wkbPolyhedralSurfaceZM    = 3015,
```

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
               wkbTinZM                    = 3016,
               }

WKBPoint {
   byte byteOrder;
   static uint32 wkbType = 1;
   Point point}

WKBPointZ {
   byte byteOrder;
   static uint32 wkbType = 1001;
   PointZ point}

WKBPointM {
   byte byteOrder;
   static uint32 wkbType = 2001;
   PointM point}

WKBPointZM {
   byte byteOrder;
   static uint32 wkbType = 3001;
   PointZM point}

WKBLineString {
   byte byteOrder;
   static uint32 wkbType = 2;
   uint32 numPoints;
   Point points[numPoints]}

WKBLineStringZ {
   byte byteOrder;
   static uint32 wkbType = 1002;
   uint32 numPoints;
   PointZ points[numPoints]}

WKBLineStringM {
   byte byteOrder;
   static uint32 wkbType = 2002;
   uint32 numPoints;
   PointM points[numPoints]}

WKBLineStringZM {
   byte byteOrder;
   static uint32 wkbType = 3002;
   uint32 numPoints;
   PointZM points[numPoints]}

WKBPolygon {
   byte byteOrder;
   static uint32 wkbType = 3;
   uint32 numRings;
   LinearRing rings[numRings]}

WKBPolygonZ {
   byte byteOrder;
   static uint32 wkbType = 1003;
   uint32 numRings;
   LinearRingZ rings[numRings]}
```

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
WKBPolygonM {
   byte byteOrder;
   static uint32 wkbType = 2003;
   uint32 numRings;
   LinearRingM rings[numRings]}

WKBPolygonZM {
   byte byteOrder;
   static uint32 wkbType = 3003;
   uint32 numRings;
   LinearRingZM rings[numRings]}

WKBTriangle {
   byte byteOrder;
   static uint32 wkbType = 17;
   uint32 numRings;
   LinearRing rings[numRings]}

WKBTriangleZ {
   byte byteOrder;
   static uint32 wkbType = 10 17;
   uint32 numRings;
   LinearRingZ rings[numRings]}

WKBTriangleM {
   byte byteOrder;
   static uint32 wkbType = 20 17;
   uint32 numRings;
   LinearRingM rings[numRings]}

WKBTriangleZM {
   byte byteOrder;
   static uint32 wkbType = 30 17;
   uint32 numRings;
   LinearRingZM rings[numRings]}

WKBPolyhedralSurface {
   byte byteOrder;
   static uint32 wkbType = 15;
   uint32 numPolygons;
   WKBPolygon polygons[numPolygons]}

WKBPolyhedralSurfaceZ {
   byte byteOrder;
   static uint32 wkbType=1015;
   uint32 numPolygons;
   WKBPolygonZ polygons[numPolygons]}

WKBPolyhedralSurfaceM {
   byte byteOrder;
   static uint32 wkbType=2015;
   uint32 numPolygons;
   WKBPolygonM polygons[numPolygons]}

WKBPolyhedralSurfaceZM {
   byte byteOrder;
   static uint32 wkbType=3015;
   uint32 numPolygons;
   WKBPolygonZM polygons[numPolygons]}
```

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
WKBTIN {
   byte byteOrder;
   static uint32 wkbType = 16;
   uint32 numPolygons;
   WKBPolygon polygons[numPolygons]}

WKBTINZ {
   byte byteOrder;
   static uint32 wkbType=1016;
   uint32 numPolygons;
   WKBPolygonZ polygons[numPolygons]}
WKBTINM {
   byte byteOrder;
   static uint32 wkbType=2016;
   uint32 numPolygons;
   WKBPolygonM polygons[numPolygons]}

WKBTINZM {
   byte byteOrder;
   static uint32 wkbType=3016;
   uint32 numPolygons;
   WKBPolygonZM polygons[numPolygons]}

WKBMultiPoint {
   byte byteOrder;
   static uint32 wkbType=4;
   uint32 numPoints;
   WKBPoint points[numPoints]}

WKBMultiPointZ {
   byte byteOrder;
   static uint32 wkbType=1004;
   uint32 numPoints;
   WKBPointZ points[numPoints]}

WKBMultiPointM {
   byte byteOrder;
   static uint32 wkbType=2004;
   uint32 numPoints;
   WKBPointM points[numPoints]}

WKBMultiPointZM {
   byte byteOrder;
   static uint32 wkbType=3004;
   uint32 numPoints;
   WKBPointZM points[numPoints]}

WKBMultiLineString {
   byte byteOrder;
   static uint32 wkbType = 5;
   uint32 numLineStrings;
   WKBLineString lineStrings[numLineStrings]}

WKBMultiLineStringZ {
   byte byteOrder;
   static uint32 wkbType = 1005;
   uint32 numLineStrings;
   WKBLineStringZ lineStrings[numLineStrings]}
```

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

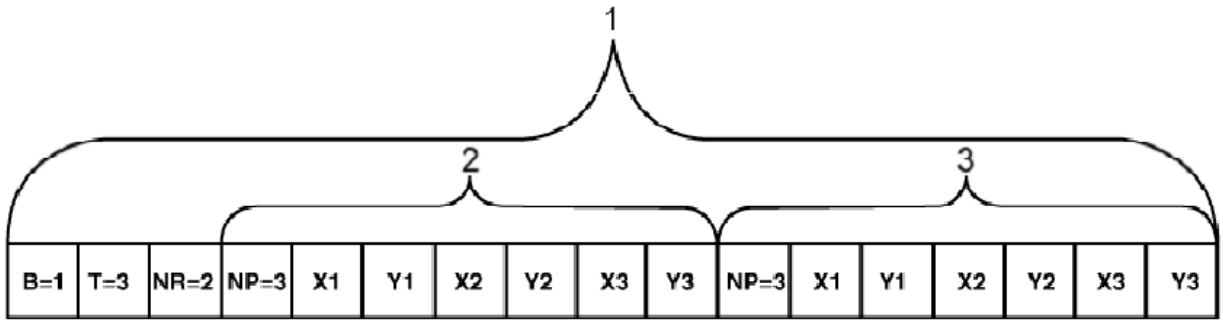
```
WKBMultiLineStringM {
   byte byteOrder;
   static uint32 wkbType = 2005;
   uint32 numLineStrings;
   WKBLineStringM lineStrings[numLineStrings]}

WKBMultiLineStringZM {
   byte byteOrder;
   static uint32 wkbType = 3005;
   uint32 numLineStrings;
   WKBLineStringZM lineStrings[numLineStrings]}

WKBMultiPolygon {
   byte byteOrder;
   static uint32 wkbType = 6;
   uint32 numPolygons;
   WKBPolygon polygons[numPolygons]}

WKBMultiPolygonZ {
   byte byteOrder;
   static uint32 wkbType = 1006;
   uint32 numPolygons;
   WKBPolygonZ polygons[numPolygons]}

WKBMultiPolygonM {
   byte byteOrder;;
   static uint32 wkbType = 2006;
   uint32 numPolygons;
   WKBPolygonM polygons[numPolygons]}

WKBMultiPolygonZM {
   byte byteOrder;
   static uint32 wkbType = 3006;
   uint32 numPolygons;
   WKBPolygonZM polygons[numPolygons]}

WKBGeometryCollection {
   byte byte_order;
   static uint32 wkbType = 7;
   uint32 numGeometries;
   WKBGeometry geometries[numGeometries]}

WKBGeometryCollectionZ {
   byte byte_order;
   static uint32 wkbType = 1007;
   uint32 numGeometries;
   WKBGeometryZ geometries[numGeometries]}

WKBGeometryCollectionM {
   byte byte_order;
   static uint32 wkbType = 2007;
   uint32 numGeometries;
   WKBGeometryM geometries[numGeometries]}

WKBGeometryCollectionZM {
   byte byte_order;
   static uint32 wkbType = 3007;
   uint32 numGeometries;
```

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
   WKBGeometryZM geometries[numGeometries]}

WKBGeometry {Union {
              WKBPoint  point;
              WKBLineString   linestring;
              WKBPolygon      polygon;
              WKBTriangle     triangle;
              WKBPolyhedralSurface  polyhedralsurface;
              WKBTIN    tin;
              WKBMultiPoint   mpoint;
              WKBMultiLineString    mlinestring;
              WKBMultiPolygon       mpolygon;
              WKBGeometryCollection       collection;
              }};

WKBGeometryZ {
              union {
              WKBPointZ       pointz;
              WKBLineStringZ  linestringz;
              WKBPolygonZ     polygonz;
              WKBTriangleZ    trianglez;
              WKBPolyhedralSurfaceZ       Polyhedralsurfacez;
              WKBTinZ   tinz;
              WKBMultiPointZ  mpointz;
              WKBMultiLineStringZ   mlinestringz;
              WKBMultiPolygonZ      mpolygonz;
              WKBGeometryCollectionZ      collectionz;
              }};

WKBGeometryM {Union {
              WKBPointM       pointm;
              WKBLineStringM  linestringm;
              WKBPolygonM     polygonm;
              WKBTriangleM    trianglem;
              WKBPolyhedralSurfaceM       Polyhedralsurfacem;
              WKBTinM   tinm;
              WKBMultiPointM  mpointm;
              WKBMultiLineStringM   mlinestringm;
              WKBMultiPolygonM      mpolygonm;
              WKBGeometryCollectionM      collectionm;
              }};

WKBGeometryZM {Union {
              WKBPointZM      pointzm;
              WKBLineStringZM       linestringzm;
              WKBPolygonZM    polygonzm;
              WKBTriangleZM   trianglezm;
              WKBPolyhedralSurfaceM       Polyhedralsurfacezm;
              WKBTinZM  tinzm;
              WKBMultiPointZM       mpointzm;
              WKBMultiLineStringZM  mlinestringzm;
              WKBMultiPolygonZ      mpolygonzm;
              WKBGeometryCollectionZM     collectionzm;
              }};
```

На Рисунке 25 показано графическое представление объекта Polygon с внешним и внутренним кольцом.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.



Key

1 WKB Polygon
2 ring 1
3 ring 2

Рисунок 25: Двоичное представление геометрического объекта в формате NDR (B = 1) типа Polygon (T = 3) с 2 LinearRings (NR = 2) каждый LinearRings состоит из 3 точек (NP = 3)

8.2.9 Утверждения для WKB-представления

WKB-представление геометрии разработано для представления экземпляров типов Geometry. Любой экземпляр `WKBGeometry` будет удовлетворять утверждениям, описанным в этой спецификации (см. п. 6.1).

9 WKT-представление пространственной системы координат

9.1 Обзор

WKT-представление пространственной системы координат предусматривает стандарт текстового представления информации о пространственной системе координат.

9.2 Описание

Пространственная система координат, или просто система координат, может быть представлена географической (широта-долгота), спроецированной (X, Y) или геоцентрической (X, Y, Z) системами координат.

Каждый объект начинается с ключевого слова, записанного в верхнем регистре (например, DATUM или UNIT), который следует после указанных в квадратных скобках и разделенных запятыми параметров объекта. Если объект состоит из нескольких объектов, результатом является вложенная структура. При реализации должна осуществляться возможность замены стандартных скобок ( ) на квадратные [ ], а так же чтение обоих видов скобок.

Приложение B содержит неисчерпывающий список систем координат и параметров используемый для описания объектов в WKT-представлениии информации о системе координат.

Определение строкового представления системы координат в форме Extended Backus Naur Form (EBNF) следует дальше, в квадратных скобках. Некоторые определения для чисел и имен взяты из Geometry WKT.

```
<spatial reference system> ::=  <projected cs>|
```

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
                                <geographic cs>|
                                <geocentric cs>
<projected cs> ::=              PROJCS <left delimiter>
                                <csname>
                                <comma> <geographic cs>
                                <comma> <projection>
                                (<comma> <parameter> )*
                                <comma> <linear unit>
                                <right delimiter>
<geographic cs> ::=             GEOGCS <left delimiter>
                                <csname>
                                <comma> <datum>
                                <comma> <prime meridian>
                                <comma> <angular unit>
                                (<comma> <linear unit>)
                                <right delimiter>
<geocentric cs> ::=             GEOCCS <left delimiter>
                                <name>
                                <comma> <datum>
                                <comma> <prime meridian>
                                <comma> <linear unit>
                                <right delimiter>
<datum> ::=                     DATUM <left delimiter> <datum
                                name>
                                <comma> <spheroid>
                                <right delimiter>
<projection> ::=                PROJECTION <left delimiter>
                                <projection name>
                                <right delimiter>
<parameter> ::=                 PARAMETER <left delimiter>
                                <parameter name>
                                <comma> <value>
                                <right delimiter>
<spheroid> ::=                  SPHEROID <left delimiter>
                                <spheroid name>
                                <comma> <semi-major axis>
                                <comma> <inverse flattening>
                                <right delimiter>
<prime meridian> ::=            PRIMEM <left delimiter>
                                <prime meridian name>
                                <comma> <longitude>
                                <right delimiter>
<linear unit> ::=               <unit>
<angular unit> ::=              <unit>
<unit> ::=                      UNIT <left delimiter>
                                <unit name>
                                <comma> <conversion factor>
                                <right delimiter>
<value> ::=                     <signed numeric literal>
<semi-major axis> ::=           <signed numeric literal>
<longitude> ::=                 <signed numeric literal>
<inverse flattening> ::=        <signed numeric literal>
<conversion factor> ::=         <signed numeric literal>
<unit name> ::=                 <quoted name>
```

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

```
<spheroid name> ::=            <quoted name>
<projection name> ::=          <quoted name>
<prime meridian name> ::=      <quoted name>
<parameter name> ::=           <quoted name>
<datum name> ::=               <quoted name>
<csname> ::=                   <quoted name>
```

Примечание: Большая полуось измеряется в метрах и должна быть больше нуля.

Примечание: Коэффициент преобразования устанавливает число метров (для линейных измерений) и ли число радиан (для угловых измерений) на единицу. Коэффициент преобразования должен быть больше нуля.

Система координат набора данных идентифицирована в ключевом слове PROJCS если данные заданы в плановых координатах, GEOGCS - в географических и GEOCCS –в геоцентрических.

Ключевое слово PROJCS следует за всеми «частями», которые заданы в плановой системе координат. Первой частью любого объекта является имя. Несколько объектов соответствуют плановой системе координат: географическая система координат, картографическая проекция, 0 или более параметров и линейные единицы измерения. Все плановые системы координат основаны на географических системах координат, поэтому специфичные для плановой системы координат части должны быть описаны в первую очередь.

ПРИМЕР 1: Описание зоны 10N UTM, датум NAD83:

```
PROJCS["NAD_1983_UTM_Zone_10N",
      <geographic cs>,
      PROJECTION["Transverse_Mercator"],
      PARAMETER["False_Easting",500000.0],
      PARAMETER["False_Northing",0.0],
      PARAMETER["Central_Meridian",-123.0],
      PARAMETER["Scale_Factor",0.9996],
      PARAMETER["Latitude_of_Origin",0.0],
      UNIT["Meter",1.0]]
```

Имя и объекты описывают географическую систему координат в порядке: датум, эллипсоид, начальный меридиан, угловые единицы измерений.

ПРИМЕР 2: Строковое описание географической системы координат зоны 10N UTM в NAD83

```
GEOGCS["GCS_North_American_1983",
      DATUM["D_North_American_1983",
      ELLIPSOID["GRS_1980",6378137,298.257222101]],
      PRIMEM["Greenwich",0],
      UNIT["Degree",0.0174532925199433]]
```

ПРИМЕР 3: Полное строковое представление зоны 10N UTM

```
PROJCS["NAD_1983_UTM_Zone_10N",
      GEOGCS["GCS_North_American_1983",
      DATUM[ "D_North_American_1983",ELLIPSOID["GRS_1980",6378137,298.257222101]],
      PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199433]],
      PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting",500000.0],
      PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",-123.0],
      PARAMETER["Scale_Factor",0.9996],PARAMETER["Latitude_of_Origin",0.0],
      UNIT["Meter",1.0]]
```

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Приложение А
(информационное)

Соответствие концепций общей архитектуры концепциям геометрической модели 19107

А.1 Введение

Это информационное приложение выявляет сходства и различия между концепциями геометрии в этой спецификации и геометрической моделью ISO 19107. Далее в приложении они упоминаются соответственно как SFA-CA и Пространственная схема.

А.2 Геометрическая модель

А.2.1 Геометрическая модель SFA-CA

Рисунок 1 показывает геометрическую модель и содержание SFA-CA. Более подробная информация содержится в 6.1.

А.2.2 Часть геометрической модели Пространственной схемы

Рисунок А.1 показывает корневой класс геометрической части Пространственной схемы. Рисунок А.2 показывает более подробно иерархию наследования. Подробная информация содержится в ISO 19107.

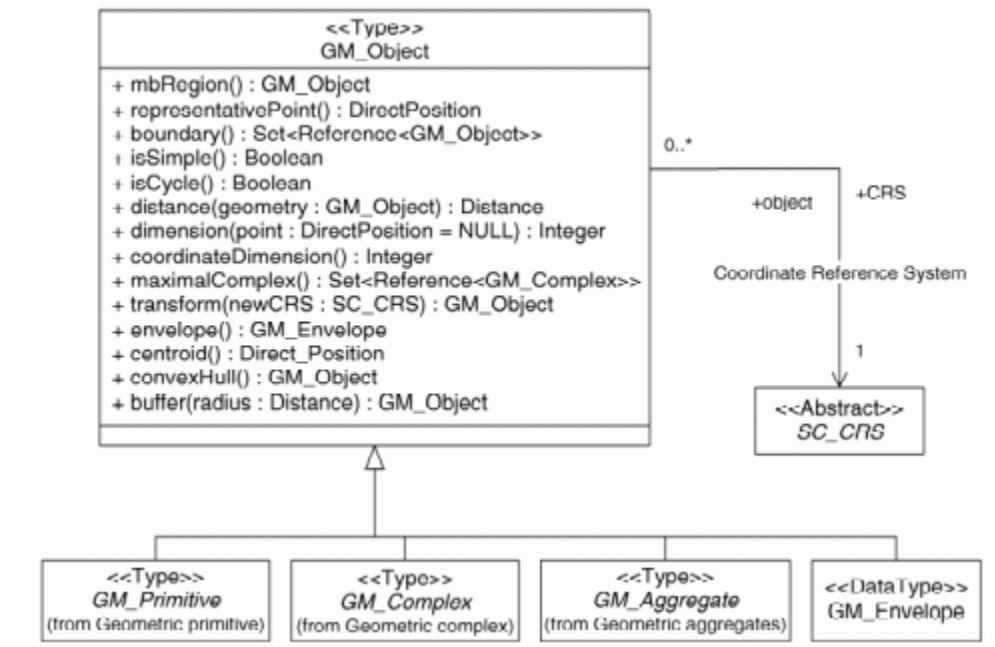


Рисунок А. 1: Корень и подчиненные Пространственной схемы

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

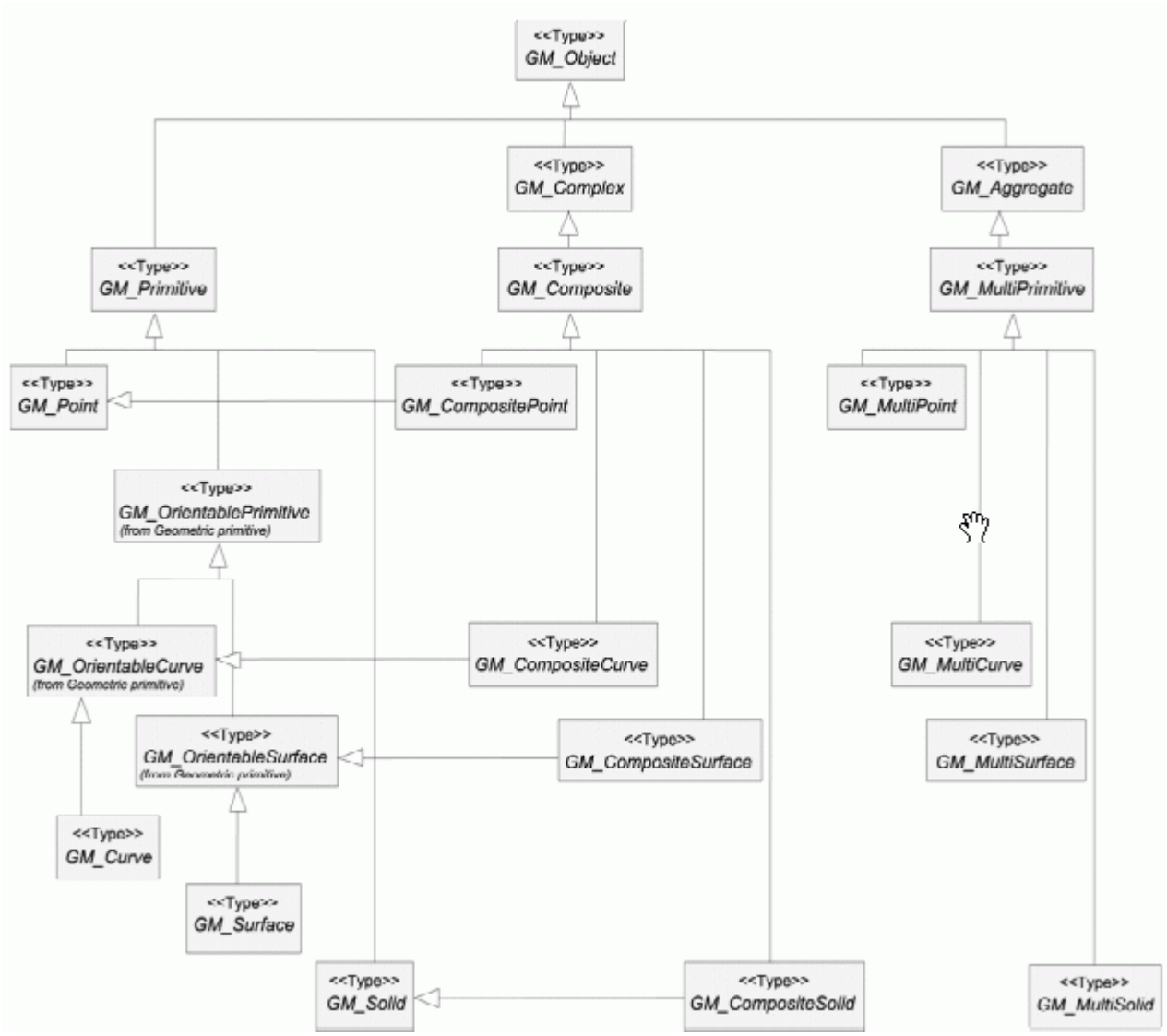


Рисунок A. 2: Иерархия GM_Object

A.3 Соответствие

A.3.1 Обзор

Концепции геометрии SFA-CA и их соответствия концепциям Пространственной схемы следующие:
- SFA-CA рассматривает только максимум 2-мерные геометрические объекты, Пространственная схема – максимум 3-мерные.
- Тип Геометрии SFA-CA соответствует GM_Object Пространственной схемы.
- Индивидуальные подтипы Типа Геометрии SFA-CA соответствуют одному или нескольким подтипам геометрической модели Пространственной схемы.
- Тип GeometryCollection SFA-CA соответствует более строгому типу GM_Aggregate Пространственной схемы.
- Концепции GM_Complex и GM_Composite Пространственной схемы показывают понятие 'manifolds'. В SFA-CA таких понятий нет.
- SFA-CA не поддерживает понятие топологии, которая явным образом моделируется топологической моделью предоставляемой Пространственной схемой.

При описании соответствий самыми важными являются второй, третий и четверый пункт. Однако, есть также заслуживающие упоминания некоторые основные принципы моделирования. Так уровень

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

абстракции между SFA-CA и Пространственной схемой важное соответствие которое можно описать следующим образом.

a) SFA-CA это реализация и спецификация зависящая от платформы;
b) Пространственная схема – абстрактна и является спецификацией не зависящей от платформы.

Таким образом, все практические соответствия, например взаимодействие, между системами основанными на спецификации SFA-CA с системами основанными только на Пространственной схеме должны учитывать конкретные представления и конкретные типы данных систем. Это особенно важно, когда сервер баз данных SFA-CA должен поддерживать несколько приложений базирующихся на Пространственной схеме.

Пример 1 Координаты x- и y в SFA-CA явным образом определяются как тип Double. В Пространственной схеме соответствующие координаты задаются как тип Number, т.е. абстрактно.

Пример 2 Все булевы операции в SFA-CA возвращают “1” если истина, и в обратном случае интерпретируются как ложь, т.е. в любом случае возвращается целочисленное значение. Подобная операция в Пространственной схеме возвращает явное булево значение.

Наконец, атрибуты в Пространственной схеме абстрактны в том, что они могут быть заданы с точки зрения операций доступа и изменения, или как конкретные атрибуты представления, в любой реализации. Детали касающиеся этих вопросов далее не рассматриваются в этом документе.

Большинство соответствий далее даются в табличной форме, например названия и описание SFA-CA показываются в первой колонке и соответствия из Пространственной схемы во второй. Если необходимо что-то подчерпнуть в соответствии, это делается в третьей колонке. Таким образом, мы подчеркиваем соответствия концепций SFA-CA концепциям Пространственной схемы, но не наоборот. В связи с этим, необходимо чтобы SFA-CA полностью содержалась в Пространственной схеме, чтобы считаться частью серии стандартов ISO 19100.

A.3.2 Тип Geometry

A.3.2.1 Обзор

В целом Тип Геометрия SFA-CA соответствует определению GM_Object в Пространственной схеме. Мы показываем все определения Типа Геометрия с соответствующими определениями типа GM_Object. Здесь мы следуем структуре этой Спецификации и разделяем описания соответствий на три части.

A.3.2.2 Основные методы геометрии

SFA-CA	Пространственная схема	Комментарий
Geometry .Dimension ( ):Integer	GM_Object::dimension(): Integer	-
Geometry.GeometryType ( ):String	Not defined	Определяется схемой приложения
Geometry.SRID ( ):Integer	GM_Object::CRS : CRS	-
Geometry.Envelope( ):Geometry	GM_Object::envelope(): GM_Envelope GM_Object::mbRegion(): GM_Object	Приложение решает какой оператор использовать
Geometry.AsText( ):String	Not defined	Определяется схемой приложения
Geometry.AsBinary( ):Binary	Not defined	Определяется схемой приложения
Geometry.IsEmpty( ):Integer	TransfiniteSet<DirectPosition>.isEmpty	Проверка на пустоту
Geometry.IsSimple( ):Integer	GM_Object::isSimple(): Boolean	-
Geometry.Boundary( ):Geometry	GM_Object::boundary(): Set<Reference<GM_Object>>	Подпись изменяется в подтипах GM_objects

A.3.2.3 Методы для проверки пространственных отношений между объектами геометрии

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

В SFA-CA, набор операторов Эгенхофера и Клементини определены прямо в типе Геометрия. Однако, в Пространственной схеме, полный набор этих операторов определен не как явные свойства поведения GM_Object, а как свободные функции для пар геометрических или топологических объектов (ISO 19107, Clause 8). GM_Object реализует операции отношений из шаблона интерфейса (параметрический классификатор ISO 19107) TransfiniteSet<DirectPosition>. Пространственные операции могут быть получены из ISO 19107 из свободных функций определенных в Clause 8: Производные топологические отношения.

SFA-CA	Пространственная схема	Комментарий
Geometry.Equals(anotherGeometry: Geometry):Integer	GM_Object::equals(pointSet: GM_Object): Boolean	-
Geometry.Intersects(anotherGeometry: Geometry):Integer	GM_Object::intersects(pointSet: GM_Object): Boolean	Оператор определения пересечения
Geometry.Contains(anotherGeometry: Geometry):Integer	GM_Object::contains(pointSet: GM_Object): Boolean	-

Для других операторов типа Геометрия, например Disjoint, Touches, Crosses, Within, Overlaps and Relate, Пространственная схема очерчивает в ISO 19107:2003 (cf. Clause 8) как определить соответствующий метод в Пространственной схеме. Отметьте, что это объяснение касается всех трёх GM_Object, GM_Primitive, и GM_Composite, как типов геометрических объектов. Тип GM_Aggregate получает эти отношения из соответствующего типа GM_Primitives, который представляет собой элемент аггрегата.

A.3.2.4 Методы, поддерживающие пространственный анализ

SFA-CA	Пространственная схема	Комментарий
Geometry.Distance(anotherGeometry: Geometry):Double	GM_Object::distance(): Distance	-
Geometry.Buffer(distance:Double): Geometry	GM_Object::buffer(radius: Distance): GM_Object	Отметьте разницу в параметрах.
Geometry.ConvexHull( ):Geometry	GM_Object::convexHull(): GM_Object	-
Geometry.Intersection(AnotherGeometry:Geometry):Geometry	GM_Object::Intersection(pointSet: GM_Object): GM_Object	В принципе этот метод используется для определения пространственных отношений выше.
Geometry.Union(anotherGeometry: Geometry):Geometry	GM_Object::union(pointSet: GM_Object): GM_Object	-
Geometry.Difference(anotherGeometry: Geometry):Geometry	GM_Object::difference(pointSet: GM_Object): GM_Object	-
Geometry.SymDifference(AnotherGeometry:Geometry):Geometry	GM_Object::symmetricDifference(pointSet: GM_Object): GM_Object	-

И SFA-CA и Пространственная схема наборы set-theoretic (i.e., set-geometric) операция, например последние четыре ряда выше, и объясняют семантику в терминах неявных point-sets. Теоретически это верно, но не явно не проверено, что point-set предпосылки верны для геометрических значений задаваемых этими двумя моделями.

A.3.3 “Атомарные” подтипы типа Геометрия

A.3.3.1 Обзор

Структура иерархии подтипов SFA-CA и Пространственной схемы сильно различаются. Однако, этот раздел покажет возможное соответствие между двумя иерархиями “атомарных” подтипов. Термин 'атомарный подтип’ относится к типу, который не является коллекцией, составным, сложным или аггрегированным. Далее мы также включим все операторы.

A.3.3.2 Точка

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

SFA-CA	Пространственная схема	Комментарий
Point	GM_Curve GM_GenericCurve GM_CurveSegment GM_LineString GM_LineSegment	Правильны обе альтернативы. DirectPosition определяет координаты, последовательность числовых координат определяет Point.
Point.X( ):Double	GM_GenericCurve::length():Length	Одно из двух, в зависимости от применения схемы
Point.Y( ):Double	GM_GenericCurve::startPoint() : DirectPosition	См. предыдущее.

A.3.3.3 Кривая

SFA-CA	Пространственная схема	Комментарий
Curve	GM_Curve GM_GenericCurve GM_CurveSegment GM_LineString GM_LineSegment	Концепция кривой в SFA-SQL может соответствовать нескольким определениям в Spatial schema.
Curve.Length( ):Double	GM_GenericCurve::length():Length	Длина определяется нескольими параметрами в зависимости от того вычисляется целое или часть кривой.
Curve.StartPoint( ):Point	GM_GenericCurve::startPoint() : DirectPosition	-
Curve.EndPoint( ):Point	GM_GenericCurve:: endPoint() : DirectPosition	-
Curve.IsClosed( ):Integer	GM_Object.isCycle() : Boolean	Задается startPoint() = endPoint(); может соответствовать GM_Object::isSimple:Boolean
Curve.IsRing( ):Integer	GM_Object.isCycle() : Boolean AND GM_Object.isSimple() : Boolean	Задается свойствами закрытости и простоты

A.3.3.4 LineString

SFA-CA	Пространственная схема	Комментарий
LineString	GM_LineString	-
LinearString.NumPoints( ):Integer	GM_LineString::controlPoints.count	Может быть получено
LinearString.PointN(N:Integer):Point	GM_LineString::controlPoints(N)	Может быть получено

A.3.3.5 LinearRing и LineSegment

Эти два типа представлены в SFA-CA как ограниченные случаи экземпляров LineString, то есть оба имеют тип LineString с дополнительными ограничениями. В SFA-CA они являются типами не создающими экземпляров, и соответственно соответствуют GM_Ring и GM_LineSegment Пространственной схемы. Отметьте однако, что SFA-CA спецификация подразумевае, что система управляет этими двумя типами подсредством добавления функциональности не определенной в SFA-SQL.

A.3.3.6 Поверхность

Тип Поверхность, стандарта SFA-CA является создающим экземпляры типом. Единственная поверхность instantiable SFA-CA планарная и простая 2D поверхность заданная типом Полигоном описываемым далее.

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

A.3.3.7 Полигон

SFA-CA	Пространственная схема	Комментарий
Polygon	GM_GenericSurface GM_Surface GM_SurfacePatch GM_Polygon	GM_Polygon и GM_SurfacePatch не показаны на Рисунке A.3, связи в этом случае более сложные см. [1].
Surface.Area( ):Double	GM_GenericSurface::area() : Area	-
Surface.Centroid( ):Point	GM_Object::centroid : DirectPosition	-
Surface.PointOnSurface( ):Point	GM_Object:: representativePoint() : DirectPosition	-
Polygon.ExteriorRing( ): LineString	GM_Polygon::exterior : GM_GenericCurve	Ввнешнее также может быть определено как 0 или более кривых в [1].
Polygon.InteriorRingN (N:Integer): LineString	*Not defined*	Может быть вычислено например из внутреннего GM_Polygon
Polygon.NumInteriorRing( ):Integer	*Not defined*	Может быть вычислено например из внутреннего GM_Polygon

A.3.3.8 PolyhedralSurface

PolyhedralSurface – непрерывная коллекция полигонов, которые разделяют общие сегменты границ и которые вместе имеют топологические свойства поверхности. PolyhedralSurface наследует все функции поверхности.

SFA-CA	Пространственная схема	Комментарий
PolyhedralSurface	GM_PolyhedralSurface as a subtype of GM_Surface	-
PolyhedralSurface.NumPatches() : Integer	GM_PolyhedralSurface.patch.count : Integer	Размер “patch”
PolyhedralSurface.PatchN (N: Integer): Polygon	GM_PolyhedralSurface.patch.getAt(N) : GM_Polygon	Получить смещение “patch”
PolyhedralSurface.BoundingPolygons (p: Polygon): MultiPolygon		Запрос “patch” для полигонов имеющих общую границу с “p”
IsClosed (): Integer	GM_Object.isCycle() : Boolean	

A.3.4 Подтипы Коллекций типа Геометрия

A.3.4.1 Обзор

Этот раздел описывает соотвествие между коллекциями в SFA-CA и аггрегатами в Пространственной схеме. Пространственная схема также предоставляет концепцию множества (manifold), с точки зрения структурированного геометрического типа представляющего коллекцию геометрических композитовто есть, каждый композит состоит из композитов на более низком уровне и измерении. Однако, эта концепция не поддерживается SFA-CA и должна реализовываться в базе данных SFA-CA другими средствами.

A.3.4.2 GeometryCollection

Это корневой тип специализированных типов коллекций, которые являются коллекциями атомарных геометрических типов определенных выше.

SFA-CA	Пространственная схема	Комментарий
GeometryCollection	GM_Aggregate and its subtype GM_MultiPrimitive	-

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

GeometryCollection :: NumGeometries( ) : Integer	GM_Aggregate.element.count : Integer	Может быть вычислен, например из роли “element” GM_Aggregate
GeometryCollection :: GeometryN(N : Integer) : Geometry	GM_Aggregate.element.getAt(N) : GM_Geometry	Может быть вычислен, например из роли “element” GM_Aggregate

Подтипы GeometryCollection, представленные ниже должны соответствовать следующим ограничениям, которые не обеспечиваются автоматически аггрегатами Пространственной схемы.

a) Для каждого элемента в GeometryCollection, его внутреннее должно быть раздельно с внутренним любого другого, отдельного элемента этой GeometryCollection.
b) Каждая граница элемента в GeometryCollection может только пересекаться границу другого, отдельного элемента в конечном количестве точек.

Далее, аггрегаты Пространственной схемы описанные ниже не определяют явных методов. Подразумевается, что методы применимые к аггрегатам как к геометрическим объектам получаются из существующих методов определенных для GM_Primitives, из которых состоят аггрегаты.

A.3.4.3 MultiPoint

SFA-CA	Пространственная схема	Комментарий
MultiPoint	GM_MultiPoint	-

MultiPoint в SFA-CA соответствует GM_MultiPoint в Пространственной схеме. Дополнительные методы для MultiPoint не определены.

A.3.4.4 MultiLineString

MultiLineString - подтип не создающего экземпляры типа MultiCurve. Отметьте использование MultiCurve в ссылках на методы MultiLineString в следующей таблице. Собственные методы для типа геометрии MultiLineString
не определены.

SFA-CA	Пространственная схема	Комментарий
MultiLineString	GM_MultiCurve GM_MultiLineString	-
MultiCurve.IsClosed( ):Integer	GM_Object.isCycle() : Boolean	Может быть получен проверкой начальной и конечной точек каждого GM_Primitive в аггрегате.
MultiCurve.Length( ):Double	GM_MultiCurve::length : Length	-

A.3.4.5 MultiPolygon

MultiPolygon – подтип не создающего экземпляры типа MultiSurface. Отметьте использование в ссылках на методы MultiPolygon в следующей таблице. Собственные методы для типа геометрии MultiPolygon.

SFA-CA	Пространственная схема	Комментарий
MultiPolygon	GM_MultiSurface	This correspondence is unclear and precaution should be taken, cf. also the correspondence for Polygon above.
MultiSurface.Area () : Double	GM_MultiSurface::area : Area	-
MultiSurface.PointOnSurface( ) :	GM_Object:: representativePoint() :	-

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Point	DirectPosition	
MultiSurface.Centroid( ):Point	GM_Object::centroid() : DirectPosition	-

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Приложение B
(информационное)

Поддерживаемые системы координат

B.1 Цель данного приложения

Это информационное приложение предоставляет неисчерпывающий список Кодов и Параметров для определения пространственной системы координат. Это приложение приводится для иллюстрации к 6.4. Это приложение может быть в будущем заменено формальным каталогом Кодов и Параметров как часть ISO 19127.

B.2 Линейные единицы

Таблица B - 1 — Линейные единицы

Имя	Значение
Metre	1,0
International Foot	0,304 8
U.S. Foot	12/39,37
Modified American Foot	12,000 458 4/39,37
Clarke's Foot	12/39,370 432
Indian Foot	12/39,370 141
Link	7,92/39,370 432
Link (Benoit)	7,92/39,370 113
Link (Sears)	7,92/39,370 147
Chain (Benoit)	792/39,370 113
Chain (Sears)	792/39,370 147
Yard (Indian)	36/39,370 141
Yard (Sears)	36/39,370 147
Fathom	1,828 8
Nautical Mile	1 852,0
South African Cape Foot	0,314 855 575 16
South African Geodetic Foot	0,304 797 265 4
German Legal Meter	1,000 013 596 5

B.3 Угловые единицы

Таблица B - 2 — Угловые единицы

Имя	Значение
Radian	1
Decimal Degree	π/180
Decimal Minute	(π/180)/60
Decimal Second	(π/180)/3 600
Gon	π/200
Grad	π/200

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

B.4 Эллипсоиды и сферы

Таблица B - 3 — Эллипсоиды и сферы

Имя	Главная полуось	Обратное уплощение
Airy	6 377 563,396	299,324 964 6
Modified Airy	6 377 340,189	299,324 964 6
Australian	6 378 160	298,25
Bessel	6 377 397,155	299,152 812 8
Modified Bessel	6 377 492,018	299,152 812 8
Bessel (Namibia)	6 377 483,865	299,152 812 8
Clarke 1866	6 378 206,4	294,978 698 2
Clarke 1866 (Michigan)	6 378 693,704	294,978 684 677
Clarke 1880 (Arc)	6 378 249,145	293,466 307 656
Clarke 1880 (Benoit)	6 378 300,79	293,466 234 571
Clarke 1880 (IGN)	6 378 249,2	293,466 02
Clarke 1880 (Modified)	6 378 249,145	293,466 315 8
Clarke 1880 (RGS)	6 378 249,145	293,465
Clarke 1880 (SGA)	6 378 249,2	293,465 98
Everest 1830	6 377 276,345	300,801 7
Everest 1975	6 377 301,243	300,801 7
Everest (Sarawak and Sabah)	6 377 298,556	300,801 7
Modified Everest 1948	6 377 304,063	300,801 7
GEM10C	6 378 137	298,257 222 101
GRS 1980	6 378 137	298,257 222 101
Helmert 1906	6 378 200	298,3
International 1924	6 378 388	297,0
Krasovsky	6 378 245	298,3
NWL9D	6 378 145	298,25
OSU_86F	6 378 136,2	298,257 22
OSU_91A	6 378 136,3	298,257 22
Plessis 1817	6 376 523	308,64
Sphere (radius = 1.0)	1	0
Sphere (radius = 6 371 000 m)	6 371 000	0
Struve 1860	6 378 297	294,73
War Office	6 378 300,583	296
WGS 1984	6 378 137	298,257 223 563

B.5 Геодезические датумы

Таблица B - 4— Геодезические датумы

Имя	Имя
Adindan	Liberia 1964
Afgooye	Lisbon
Agadez	Loma Quintana
Australian Geodetic Datum 1966	Lome
Australian Geodetic Datum 1984	Luzon 1911
Ain el Abd 1970	Mahe 1971
Amersfoort	Makassar
Aratu	Malongo 1987
Arc 1950	Manoca
Arc 1960	Massawa
Ancienne Triangulation Française	Merchich
Barbados	Militar-Geographische Institute
Batavia	Mhast

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Beduaram	Minna
Beijing 1954	Monte Mario
Reseau National Belge 1950	M'poraloko
Reseau National Belge 1972	NAD Michigan
Bermuda 1957	North American Datum 1927
Bern 1898	North American Datum 1983
Bern 1938	Nahrwan 1967
Bogota	Naparima 1972
Bukit Rimpah	Nord de Guerre
Camacupa	NGO 1948
Campo Inchauspe	Nord Sahara 1959
Cape	NSWC 9Z-2
Carthage	Nouvelle Triangulation Française
Chua	New Zealand Geodetic Datum 1949
Conakry 1905	OS (SN) 1980
Corrego Alegre	OSGB 1936
Côte d'Ivoire	OSGB 1970 (SN)
Datum 73	Padang 1884
Deir ez Zor	Palestine 1923
Deutsche Hauptdreiecksnetz	Pointe Noire
Douala	Provisional South American Datum 1956
European Datum 1950	Pulkovo 1942
European Datum 1987	Qatar
Egypt 1907	Qatar 1948
European Reference System 1989	Qornoq
Fahud	RT38
Gandajika 1970	South American Datum 1969
Garoua	Sapper Hill 1943
Geocentric Datum of Australia 1994	Schwarzeck
Guyane Française	Segora
Hartebeeshoek(WGS84) South African	Serindung
Herat North	Stockholm 1938
Hito XVIII 1963	Sudan
Hu Tzu Shan	Tananarive 1925
Hungarian Datum 1972	Timbalai 1948
Indian 1954	TM65
Indian 1975	TM75
Indonesian Datum 1974	Tokyo
Jamaica 1875	Trinidad 1903
Jamaica 1969	Trucial Coast 1948
Japanese Geodetic Datum 2000	Voirol 1875
Kalianpur	Voirol Unifie 1960
Kandawala	WGS 1972
Kertau	WGS 1972 Transit Broadcast Ephemeris
Kuwait Oil Company	WGS 1984
La Canoa	Yacare
Lake	Yoff
Leigon	Zanderij

B.6 Главные меридианы

Table B - 5 — Главные меридианы

Имя	Значение
Greenwich	0° 0' 0"

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Bern	7° 26' 22,5" E
Bogota	74° 4' 51,3" W
Brussels	4° 22' 4,71" E
Ferro	17° 40' 0" W
Jakarta	106° 48' 27,79" E
Lisbon	9° 7' 54,862" W
Madrid	3° 41' 16,58" W
Paris	2° 20' 14,025" E
Rome	12° 27' 8,4" E
Stockholm	18° 3' 29" E

B.7 Проекции

Таблица B - 6 — Проекции

Цилиндрические проекции	Конические проекции
Cassini	Albers conic equal-area
Gauss-Kruger	Lambert conformal conic
Mercator	Azimuthal or Planar Projections
Oblique Mercator (Hotine)	Polar Stereographic
Transverse Mercator	Stereographic

B.8 Параметры проекций

Таблица B - 8 — Параметры проекций

Имя	Описание
central_meridian	Долгота выбранная за начало отсчета для координаты x
scale_factor	Множитель для уменьшения расстояния полученного с карты до актуального расстояния датума карты
standard_parallel_1	Широта вдоль которой нет искажений расстояний. Так же называется ‘широта истинного масштаба’
standard_parallel_2	Широта вдоль которой нет искажений расстояний.
longitude_of_center	Долгота центральной точки проекции
latitude_of_center	Широта центральной точки проекции
latitude_of_origin	Широта координаты y начала отсчета
false_easting	Смещение координаты x; используется, чтобы получить положительные значения
false_northing	Смещение координаты y; используется, чтобы получить положительные значения
azimuth	Угол на восток от севера определяющий центральную линию для косых проекций
longitude_of_point_1	Долгота первой точки необходимой для проекции
latitude_of_point_1	Широта первой точки необходимой для проекции
longitude_of_point_2	Долгота второй точки необходимой для проекции
latitude_of_point_2	Широта второй точки необходимой для проекции

Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.

Библиография

[1] *The OpenGIS Abstract Specification: An Object Model for Interoperable Geoprocessing*, Revision 1, OpenGIS Consortium, Inc, OpenGIS Project Document Number 96-015R1, 1996

[2] *OpenGIS Project Document 96-025: Geodetic Reference Systems*, OpenGIS Consortium, Inc., October 14, 1996

[3] Petrotechnical Open Software Consortium (POSC) *Epicentre Model*, available at: <ftp://posc.org/Epicentre/>, July 1995

[4] CLEMENTINI, E., DI FELICE, P., VAN OOSTROM, P. *A Small Set of Formal Topological Relationships Suitable for End-User Interaction, in D. Abel and B. C. Ooi (Ed.), Advances in Spatial Databases* — Third International Symposium. SSD 1993. LNCS **692**, pp. 277-295. Springer Verlag. Singapore (1993)

[5] CLEMENTINI E. AND DI FELICE P. *A Comparison of Methods for Representing Topological Relationships*, Information Sciences **80** (1994), pp. 1-34

[6] CLEMENTINI, E. AND DI FELICE, P. A *Model for Representing Topological Relationships Between Complex Geometric Features in Spatial Databases*, Information Sciences **90(1-4)** (1996), pp. 121-136

[7] CLEMENTINI E., DI FELICE P AND CALIFANO, G. *Composite Regions in Topological Queries, Information Systems*, **20(6)** (1995), pp. 33-48

[8] EGENHOFER, M.J. AND FRANZOSA, R. *Point Set Topological Spatial Relations*, International Journal of Geographical Information Systems, **5(2)** (1991), pp. 161-174

[9] EGENHOFER, M.J., CLEMENTINI, E. AND DI FELICE, P. *Topological relations between regions with holes*, International Journal of Geographical Information Systems, **8(2)** (1994), pp. 129-142

[10] EGENHOFER, M.J. AND HERRING, J. *A mathematical framework for the definition of topological relationships*. Proceedings of the Fourth International Symposium on Spatial Data Handling, Columbus, OH, pp. 803-813

[11] EGENHOFER, M.J. AND HERRING, J. *Categorizing binary topological relationships between regions, lines and points in geographic databases*, Tech. Report 91-7, National Center for Geographic Information and Analysis, Santa Barbara, CA (1991)

[12] EGENHOFER, M.J. AND SHARMA, J. *Topological Relations between regions in* $_2$ *and* Z_2, Advances in Spatial Databases — Third International Symposium, SSD 1993, **692**, Lecture Notes in Computer Science, pp. 36-52, Springer Verlag, Singapore (1993)

[13] WORBOYS, M.F. AND BOFAKOS, P. *A Canonical model for a class of areal spatial objects*, Advances in Spatial Databases — Third International Symposium, SSD 1993, **692**, Lecture Notes in Computer Science, pp. 36-52, Springer Verlag, Singapore (1993).

[14] WORBOYS, M.F. *A generic model for planar geographical objects*, International Journal of Geographical Information Systems (1992) **6(5)**, pp. 353-372

[15] *CORBA services: Common Object Services Specification*, Ch 8. Externalization Service Specification, OMG. Available at <http://www.omg.org/technology/documents/corba_spec_catalog.htm>

[16] *Distributed Component Object Model* — DCOM 1.0, Microsoft Corporation. Available at <http://www.microsoft.com/com/tech/DCOM.asp>

[17] ISO 19101:2002, *Geographic information — Reference model*

[18] IEEE 754, *IEEE Standard for binary Floating-Point Arithmetic*

 Copyright © 2006 Open Geospatial Consortium, Inc. Все права защищены.