Geometrías

Índice de contenido

1 Geometrías. Introducción

Esta interfaz es equivalente a la GM_Object especificado en la norma ISO 19107. Es la clase raíz de la taxonomía objeto geométrico y soporta interfaces comunes a todos los objetos referenciados geográficamente geométricas.

Geometría son conjuntos de posiciones directas en un sistema de coordenadas de referencia particular. Una geometría puede ser considerado como un conjunto infinito de puntos que satisface las interfaces de operación de ajuste para un conjunto de posiciones directas.

Un objeto geométrico será una combinación de una geometría de coordenadas y un sistema de coordenadas de referencia. En todas las operaciones, todos los cálculos geométricos se hace en el sistema de coordenadas de referencia del primer objeto geométrico visitada, que normalmente es el objeto cuya operación está siendo invocado. Objetos devueltos deberán estar en el sistema de coordenadas de referencia en el que los cálculos se hacen a menos que se especifique lo contrario.

2 El concepto de tipo y subtipo

En gvSIG, el tipo únicamente hace referencia a la clase de objeto geomético dependiendo de sus características geométicas, pero no indica si la geometría es en 2 dimensiones o en 3 dimensiones, o si soporta coordenada M o no la soporta. Para ello se ha creado el concepto de subtipo que sirve para indicar la dimensión de la geometría.

Los tipos de geometrías actualmente soportados por gvSIG se encuentran definidos en TYPES y son:

- Geometrías simples
  - AGGREGATE: Un conjunto que puede contener Puntos, líneas y polígonos
  - LINE | CURVE: Una geometría recta o curva continua e indefinida de puntos. Puede usarse indistintamente la constante LINE o CURVE, representan el mismo tipo de geometría.
  - POINT: Figura geométrica adimensional: no tiene longitud, área, volumen, ni otro ángulo dimensional. Describe una posición en el espacio, determinada respecto de un sistema de coordenadas preestablecido.
  - POLYGON | SURFACE: Figura plana compuesta por una secuencia finita de segmentos rectos consecutivos que cierran una región en el espacio. Puede usarse indistintamente la constante LINE o CURVE, representan el mismo tipo de geometría.
- Geometrías multiples
  - MULTICURVE: Conjunto de lineas (ya sean curvas o rectas)
  - MULTIPOINT: Conjunto de puntos
  - MULTISURFACE: Conjunto de superficies

Los subtipos de gvSIG se encuentran definidos en SUBTYPES y son

- D2 (GEOM2D): Geometrías de 2 dimensiones
- D2M (GEOM2DM): Geometrías de 2 dimensiones y con coordenada M
- D3 (GEOM3D): Geometrías de 3 dimensiones

- D3M (GEOM3DM): Geometrías de 3 dimensiones y con coordenada M

3 Creación de geometrías

Dentro del entorno de scripting en gvSIG, se ha creado un módulo, geom.py, que simplifica algunas de las operaciones más habituales con las geometrías.

Para crear una geometría debemos importar en nuestro script el módulo 'geom' e invocar a la función *createGeometry* indicando el tipo y el subtipo de la nueva geometría.

```
from geom import *


def main():
  geometry = createGeometry(POINT, D2)
```

La geometría así creada es una geometría de tipo *punto* y subtipo *2d* vacía.

4 Geometría de tipo Punto. Función createPoint

Podemos crear una geometría de tipo punto (POINT) mediante la función *createPoint* del módulo geom que devuelve una geometría y cuya sintaxis es:

```
createPoint([x, y][, subtype])
```

- createPoint(x, y, subtype)
  - x, double (opcional): Valor de la coordenada X del punto. El valor por defecto es 0.0
  - y, double (opcional): Valor de la coordenada Y del punto. El valor por defecto es 0.0
  - subtype, int (opcional): Indica las dimensiones de la geometría y si soporta coordenada z. El valor por defecto es D2

Los métodos más habituales de una geometría de tipo punto son:

- getCoordinateAt(dimension): Obtine el valor de la coordenada en una dimensión concreta.
  - dimension, int: Dimensión de la que obtener la coordenada
- getX(): Devuelve el valor de la coordenada X
- getY(): Devuelve el valor de la coordenada Y
- setCoordinateAt(dimension, value): Establece el valor de una dimensión concreta
  - dimension, int: Dimensión sobre la que se establece la coordenada
  - value, double: Valor a asignar a la coordenada
- setX(x): Establece el valor de la coordenada X
  - x, double: Valor a asignar a la coordenada X
- setY(y): Establece el valor de la coordenada Y
  - y, double: Valor a asignar a la coordenada Y

Supongamos que tenemos una geometría de tipo punto y queremos obtener sus coordenadas "x" e "y" para poder crear otra geometría de tipo pùnto con esos valores.

```
from geom import *


def main():
  #
```

```
#codigo ...

x = point.getX()
y = point.getY()

newPoint = createPoint(x,y)
```

Para editar o recuperar el resto de valores de las otras dimensiones hay que utilizar los métodos genéricos *setCoordinateAt* y *getCoordinateAt*.

Por ejemplo, para crear un punto en 3 dimensiones cuyas coordenadas sean 1,1,5 hay que ejecutar el siguiente código:

```
point = createGeometry(POINT, D3)
point.setCoordinateAt(0, 1)
point.setCoordinateAt(1, 1)
point.setCoordinateAt(2, 5)
```

Otra opción podría ser hacerlo utilizando los métodos setX() y setY() de la geometría y el método setCoordinateAt(dimension, value) y el punto obtenido sería el mismo:

```
point = createGeometry(POINT, D3)
point.setX(1)
point.setY(1)
point.setCoordinateAt(2, 5)
```

Para obtener información más completa sobre los métodos de una geometría de tipo punto, puede consultarse la documentación de la geometría de tipo Punto

5 Geometrías de tipo Polygon. Función createPolygon

La función *createPolygon* del módulo *geom*, nos permite crear una geometría de tipo polígono (POLYGON) de forma rápida.

La sintaxis de la función es

```
createPolygon([subtype])
```

- createPolygon(subtype): Crea una geometría de tipo polígono.
  - subtype, int (opcional): Indica las dimensiones de la geometría y si soporta coordenada z. El valor por defecto es D2

Los métodos comunes de estas geometrías son:

- addVertex(point): Añade un vértice a la geometría
  - point, Point: Geometría de tipo punto a añadir al polígono
- closePrimitive(): Cierra la geometría
- getBounds(): Rectangle Devuelve el límite rectangular de la geometría.
- getCoordinateAt(int index, int dimension): Obtiene el valor de la coordenada de una dimensión concreta
- getEnvelope(): Devuleve el bounding box mínimo de la geometría
- getNumVertices(): Devuelve el número de vértices de la geometría
- getVertex(int index): Devuelve una geometría de tipo punto del vértice *index* de la

geometría

Un ejemplo, el siguiente código crea una geometría de tipo polígono, a la que añade 4 vértices.

```
from geom import *

    def main():
    polygon = createPolygon(D2M)

    x = (1, 1, 5, 5)
    y = (1, 5, 5, 1)
    points = [createPoint(p[0], p[1]) for p in zip(x, y)]
    for point in points:
        polygon.addVertex(point)

    polygon.closePrimitive()
```

6 Geometría de tipo Línea. Función createLine

La función *createLine* del módulo *geom*, nos permite crear una geometría de tipo línea de forma rápida.

La sintaxis de la función es

```
createLine([subtype])
```

- createLine(subtype): Crea una geometría de tipo línea (LINE).
    - subtype, int (opcional): Indica las dimensiones de la geometría y si soporta coordenada z. El valor por defecto es D2

Los métodos comunes de estas geometrías son:

- addVertex(point): Añade un vértice a la geometría
    - point, Point: Geometría de tipo punto a añadir al polígono
- closePrimitive(): Cierra la geometría
- getBounds(): Rectangle Devuelve el límite rectangular de la geometría.
- getCoordinateAt(int index, int dimension): Obtiene el valor de la coordenada de una dimensión concreta
- getEnvelope(): Devuleve el bounding box mínimo de la geometría
- getNumVertices(): Devuelve el número de vértices de la geometría
- getVertex(int index): Devuelve una geometría de tipo punto del vértice *index* de la geometrías

7 Geometría de tipo Multipunto. Función createMultiPoint

Las geometrías MULTIPOINT, son geometrías que contienen a su vez una colección de geometrías de tipo POINT.

Podemos crear una geometría de tipo MULTIPOINT invocando a la función *createMultiPoint*, indicando el subtipo de geometría que queremos crear.

```
createMultiPoint([subtype])
```

- createMultiPoint(subtype): Crea una geometría de tipo MULTIPOINT
  - subtype, int: Subtipo de la geometría

El siguiente código crea una geometría de tipo MULTIPOINT y añade 6 geometrías de tipo POINT

```
from geom import *

def main():
  multipoint = createMultiPoint(subtype=D2)
  x = (1, 1, 5, 5, 10, 0)
  y = (1, 5, 5, 1, 10, 10)
  points = [createPoint(p[0], p[1]) for p in zip(x, y)]
  for point in points:
    multipoint.addPoint(point)
```

También pueden añadirse puntos a la geometría MULTIPOINT pasando como parámetros dos listas, una con las coordenadas X y otra con las coordenadas Y de los puntos que van a formar la multigeometría.

```
from geom import *

def main():
  multipoint = createMultiPoint(subtype=D2)
  x = (1, 1, 5, 5, 10, 0)
  y = (1, 5, 5, 1, 10, 10)
  multipoint.setPoints(x, y)
```

Los métodos de la clase MULTIPOINT son:

```
- addPoint(point)
- cloneGeometry()
- getEnvelope()
- getPrimitivesNumber()
- setPoints(xCoords, yCoords)
- getPointAt(index)
```

- addPoint(Point point): Añade una geometría de tipo POINT a la multigeometría
  - Point point: punto a añadir en la geometría
- cloneGeometry(): Crea una copia de esta geometría.
- getEnvelope(): Devuleve la mínima envolvente de la geometría.
- getPrimitivesNumber(): Devuleve el número de geometrías que forman la multigeometría
- setPoints(double[] xCoords, double[] yCoords): Inserta en la multigeometría los puntos correspondientes.
  - double[] xCoords: Lista de coordenadas X de los puntos
  - double[] yCoords: Lista de coordenadas Y de los puntos
- getPointAt(int index): devuelve el punto de una posición concreta de la multigeometría
  - int index: posición del punto

8 Geometría de tipo Multilínea. Función createMultiLine

Las geometrías MULTILINE, son geometrías que contienen a su vez una colección de geometrías de tipo LINE.

Podemos crear una geometría de tipo MULTILINE invocando a la función *createMultiLine*, indicando el subtipo de geometría que queremos crear.

```
createMultiLine([subtype])
```

- createMultiLine(subtype): Crea una geometría de tipo MULTILINE
  - subtype, int: Subtipo de la geometría

Los métodos de la clase MULTILINE son:

```
- addCurve(line)
- getCurveAt(index)
- cloneGeometry()
- getEnvelope()
- getPrimitivesNumber()
```

- addCurve(line): Añade una nueva línea (LINE) a la multigeometría
  - LINE line: Línea que se añade
- getCurveAt(index): Devuleve una línea LINE de una posición concreta de la multigeometría
  - int index: Posición de la línea dentro de la multigeometría
- cloneGeometry(): Crea una copia de esta geometría.
- getEnvelope(): Devuleve la mínima envolvente de la geometría.
- getPrimitivesNumber(): Devuelve el número de geometrías que forman la multigeometría

9 Geometría de tipo Multipolígono. Función createMultiPolygon

Las geometrías MULTIPOLYGON, son geometrías que contienen a su vez una colección de geometrías de tipo POLYGON.

Podemos crear una geometría de tipo MULTIPOLYGON invocando a la función *createMultiPolygon*, indicando el subtipo de geometría que queremos crear.

```
createMultiPolygon([subtype])
```

- createMultiPolygon(subtype): Crea una geometría de tipo MULTILINE
  - subtype, int: Subtipo de la geometría

Los métodos de la clase MULTIPOLYGON son:

```
- addSurface(polygon)
- cloneGeometry()
- getSurfaceAt(index)
- cloneGeometry()
- getEnvelope()
```

```
- getPrimitivesNumber()
```

- addSurface(polygon): Añade un nuevo polígono (POLYGON) a la multigeometría
    - POLYGON polygon: Polígono que se añade
- getSurfaceAt(index): Devuelve un polígono (POLYGON) de una posición concreta de la multigeometría
    - int index: Posición de la línea dentro de la multigeometría
- cloneGeometry(): Crea una copia de esta geometría.
- getEnvelope(): Devuleve la mínima envolvente de la geometría.
- getPrimitivesNumber(): Devuelve el número de geometrías que forman la multigeometría

10 Operaciones con geometrías

- area(): Devuelve el área de la geometría
- centroid(): Devuelve el centroide de la geometría
- perimeter(): Devuelve el perímetro de la geometría
- getEnvelope(): Devuelve el Bounding Box mínimo de la geometría
- buffer(distance): Devuelve la geometría resultante de calcular un área alrededor de la geometría utilzando la distancia del parámetro.
    - distance, double: radio del buffer.
- convexHull(): Devuelve la envolvente convexa de la geometría
- distance(geometry): Devuelve la distancia mínima entre la geometría y la geometría del parámetro.
    - geometry, Geometry: Geometría distante
- intersection(other): Devulve la geometría resultante de la intersección
    - other, Geometry: Geometría sobre la que se busca la intersección
- intersects(geometry): Devuleve True si ambas geometrías intersectan
    - geometry, Geometry: Geometría a comprobar
- closestPoints(geometry): Devuelve los puntos más cercanos de las respectivas geometrías
    - geometry[], Geometry: Puntos (POINT) más cercanos de cada una de las geometrías

Para más información puede consultarse el API de geometrías de gvSIG

gvSIG Association

Plaza Don Juan de Villarrasa 14-5,

46001, Valencia (Spain)

Registro Nacional de Asociaciones: 596206

e-mail : info@gvsig.com

Web: www.gvsig.com

Web del proyecto: http://www.gvsig.org

Documentación realizada por Víctor Acevedo.

Listas de Distribución

Existen tres listas de distribución con el objeto de facilitar la comunicación entre todos los interesados en el proyecto gvSIG. Las dos primeras, la de usuarios y la de desarrolladores, están principalmente orientadas a la comunidad de habla hispana, siendo el castellano el idioma preferente a utilizar en las mismas. La tercera de ellas, lista internacional, está orientada principalmente al resto de comunidades y la lengua preferente a utilizar es la inglesa.

- **Lista de usuarios**. Aquí podéis hacer llegar vuestra opinión sobre el funcionamiento: qué cosas os gustaría que se desarrollaran, dudas en el uso de gvSIG y todo aquello que penséis que tiene cabida en una lista de usuarios. El enlace para la suscripción a la lista de usuarios es:
  http://listserv.gva.es/mailman/listinfo/gvsig_usuarios

- **Lista de desarrolladores**. Está orientada para todos los interesados en conocer cómo está desarrollado el gvSIG. El enlace para la suscripción a esta lista esa:
  http://listserv.gva.es/mailman/listinfo/gvsig_desarrolladores

- **Lista internacional**. Está orientada tanto para usuarios como para desarrolladores de habla no hispana. El idioma a utilizar es preferentemente inglés. El enlace para la suscripción a esta lista es:
  http://listserv.gva.es/mailman/listinfo/gvsig_internacional