

**УТВЕРЖДЕН**  
**ПАРБ.00167-01 33 01-ЛУ**

**ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ  
ГЕОИНФОРМАЦИОННЫХ WEB-ПРИЛОЖЕНИЙ  
GIS WebToolKit SE**

**Руководство программиста**

**ПАРБ.00167-01 33 01**

**Листов 71**

ИНВ. № ПОДЛ	ПОДЛ И ЛАТА	ВЗАМ. ИНВ. №	ИНВ. № ЛУБЛ	ПОДЛ И ЛАТА

**2015**

## **АННОТАЦИЯ**

В данном документе содержатся сведения для обеспечения функционирования и применения инструментальных средств разработки геоинформационных Web-приложений GIS WebToolKit SE.

GIS WebToolKit SE предназначен для встраивания компонентов доступа к пространственным данным и отображения цифровой информации о местности в Web-приложениях. Инструментарий обеспечивает получение атрибутивных характеристик объектов карты по запросам, поиск объектов с заданными атрибутами или поиск на основе пространственного фильтра. Выполняются расчеты геометрических свойств, топологические оверлейные операции на векторных моделях пространственных объектов. Имеется набор программных средств редактирования пространственных объектов через Web, получения данных для пространственного моделирования рельефа местности, его анализа и изучения.

Инструментальные средства разработки GIS WebToolKit SE представляют собой набор JavaScript-классов для выполнения запросов к Web-сервису GIS WebService SE и компонентов интерактивной карты.

GIS WebService SE поддерживает протоколы обмена пространственными данными по стандартам Open Geospatial Consortium (OGC): Web Map Tile Service (OGC WMTS), Web Map Service (OGC WMS), Web Feature Service (OGC WFS), Web Coverage Service (OGC WCS) и дополнительные расширенные запросы данных на основе REST API.

В данном документе описаны методы и объекты программного интерфейса для разработчиков геоинформационных Web-приложений на языке JavaScript.

## СОДЕРЖАНИЕ

<b>1 Назначение и условия применения программы .....</b>	<b>5</b>
1.1 Назначение программы .....	5
1.2 Условия применения программы .....	5
1.2.1 Программные требования .....	5
1.2.2 Системные требования .....	6
<b>2 Характеристики программы .....</b>	<b>7</b>
<b>3 Обращение к программе.....</b>	<b>8</b>
3.1 Класс WmtsQueries .....	8
3.1.1 Методы класса WmtsQueries.....	8
3.1.2 Обработчики событий при выполнении запросов к серверу.....	9
3.1.3 Параметры запроса рисунков тайлов - аргумент <i>querytile</i> .....	9
3.1.4 Параметры запроса информации об объекте - аргумент <i>queryfeature</i> .....	10
3.2 Класс WmsQueries .....	10
3.2.1 Методы класса WmsQueries .....	11
3.2.2 Параметры запроса изображения карты - аргумент <i>querymap</i> .....	11
3.2.3 Параметры запроса информации об объекте – аргумент <i>queryfeature</i> .....	12
3.3 Класс WfsQueries .....	12
3.3.1 Методы класса WfsQueries.....	13
3.3.2 Обработчики событий при выполнении запросов к серверу .....	15
3.3.3 Параметры запроса объекта - аргумент <i>queryfeature</i> .....	16
3.4 Класс WcsQueries.....	17
3.4.1 Методы класса WcsQueries .....	17
3.4.2 Параметры запроса поверхности - аргумент <i>coverageoptions</i> .....	17
3.5 Выполнение специальных расчетов по карте .....	18
3.5.1 Методы класса MapMath.....	18
3.5.2 Ошибки выполнения расчетов.....	21
3.6 Компонент Карта Map .....	21
3.6.1 Параметры карты .....	22
3.6.2 Методы управления состоянием карты .....	23
3.6.3 Методы получения информации о карте.....	23
3.6.4 Методы управления слоями карты.....	23
3.6.5 Методы назначения обработчиков событий .....	24
3.6.6 Методы отбора объектов карты.....	24
3.6.7 Методы настройки интерфейса пользователя.....	24
3.6.8 Методы управления оверлеями карты .....	24
3.6.9 События класса .....	24
3.7 Компонент Слой карты Layer .....	25
3.7.1 Параметры слоя карты.....	25
3.7.2 Методы класса Layer .....	26
3.7.3 Подключение внешних геопорталов в качестве слоев карты .....	26
3.8 Компонент WMS-слой карты WmsLayer .....	27
3.8.1 Методы класса WmsLayer .....	27
3.9 Компонент Состав карты MapContentControl .....	28
3.9.1 Методы класса MapContentControl .....	28
3.9.2 Описание структуры дерева компонента MapContentControl .....	30
3.10 Компонент текстового поиска объектов карты mapTextSearch .....	30
3.10.1 Методы класса mapTextSearch.....	31
3.11 Компонент Панель поиска TextSearchControl.....	31

3.11.1 Методы класса TextSearchControl .....	31
3.12 Компонент метка placemark.....	31
3.12.1 Методы класса placemark .....	32
3.13 Компонент Редактор карты mapeditor .....	32
3.13.1 Описание параметров инициализации объекта mapeditor .....	33
3.13.2 Пример использования компонента mapeditor.....	33
3.14 Компонент polygon .....	35
3.14.1 Методы класса polygon.....	35
3.15 Компонент sliderToolsControl .....	36
3.15.1 Методы класса sliderToolsControl .....	36
3.15.2 События класса sliderToolsControl .....	37
3.15.3 Пример использования компонента sliderToolsControl .....	38
3.16 Компонент Маршрутизация и анализ данных с БПЛА routeBPLA .....	39
3.16.1 Описание параметров инициализации объекта routeBPLA .....	40
3.16.2 Методы класса routeBPLA .....	40
3.16.3 События класса routeBPLA .....	42
3.16.4 Пример использования компонента routeBPLA .....	43
3.17 Компонент Отображение сведений государственного кадастра недвижимости rosreestr.....	45
3.17.1 Пример использования компонента rosreestr .....	45
3.17.2 Компонент rosreestrControl .....	46
<b>4 Входные и выходные данные.....</b>	<b>49</b>
<b>5 Сообщения программы.....</b>	<b>50</b>
5.1 Сообщения класса WmtsQueries.....	50
<b>6 Проверка программы.....</b>	<b>51</b>
6.1 Предмет проверки.....	51
6.2 Порядок проверки.....	51
6.2.1 Порядок проверки выполнения операций протокола OGC WMTS .....	51
6.2.2 Порядок проверки выполнения операций протокола OGC WFS.....	54
6.2.3 Порядок проверки выполнения операций протокола OGC WMS .....	61
6.2.4 Порядок проверки выполнения операций протокола OGC WCS .....	64
<b>7 Применение программы.....</b>	<b>68</b>
7.1 Порядок использования .....	68
7.1.1 Подключение библиотек скриптов и файла стилей .....	68
7.1.2 Настройка параметров работы компонента карты .....	68
7.1.3 Создание экземпляра компонента карты.....	69
7.2 Пользовательский интерфейс .....	69
7.2.1 Перемещение изображения карты.....	70
7.2.2 Изменение масштаба отображения карты .....	70
7.2.3 Изменение состава отображаемых слоев карты .....	70
7.2.4 Получение информации об объектах карты.....	71

## 1 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ

### 1.1 Назначение программы

GIS WebToolKit SE представляет собой библиотеку компонентов и классов на языке JavaScript для встраивания компонентов доступа к пространственным данным, выполнения специальных расчетов и оверлейных операций, отображения цифровой информации о местности в Web.

Классы GIS WebToolKit SE используются для обмена данными с картографическим сервером без применения элементов интерфейса пользователя.

Компоненты реализованы в виде классов, функций и элементов управления HTML, встраиваемых в Web-страницу.

GIS WebToolKit SE использует сервис карт GIS WebService SE как источник геопространственных данных для получения графических изображений карты, геометрии объектов и различной справочной информации об объектах карты, получения пространственной информации о рельефе местности, выполнения расчетов и измерений по карте. Обмен данными с сервисом GIS WebService SE выполняется по протоколу HTTP.

Средства выбора и поиска объектов GIS WebToolKit SE позволяют получать характеристики объектов в указанной точке местности и наоборот, получать данные объектов с заданными атрибутами.

Средства пространственного анализа содержат функции поиска объектов по заданной области, функции оверлейных операций над пространственными объектами, расчеты геометрических свойств объектов.

Оверлейные операции над пространственными объектами содержат методы построения пересечений линейного и площадного объектов или двух площадных, методы построения объединения пересекающихся площадных объектов, построения линий сплайнов различных типов.

Средства специальных расчетов содержат функции определения длины, площади, периметра объектов, расстояния между объектами с учетом рельефа местности, расчета азимута, дирекционного угла.

Средства редактирования пространственных объектов включают функции создания, удаления и изменения пространственных объектов. Редактирование данных выполняется через операции транзакций протокола WFS-T.

Получение данных для пространственного моделирования рельефа местности и дальнейшего анализа полученных моделей выполняются через запросы по спецификациям OGC WCS, WFS, WMS.

Компоненты и классы GIS WebToolKit SE размещаются на Web-сервере и доступны для использования сразу после их загрузки на страницу в браузере.

### 1.2 Условия применения программы

#### 1.2.1 Программные требования

Для работы инструментальных средств разработки геоинформационных Web-приложений GIS WebToolKit SE необходимо установить на Web-сервере и настроить для использования картографический сервис GIS WebService SE.

Для подключения GIS WebToolKit SE необходимо, чтобы компоненты были загружены в браузер вместе с кодом страницы как обычные внешние JavaScript-файлы.

Наиболее распространенным способом подключения внешних скриптов является использование элемента Script в заголовке HTML-документа.

Например,

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
```

```
<title></title>
<script src="Scripts/map.js" type="text/javascript" ></script>
<script src="Scripts/wqueries.js" type="text/javascript" ></script>
<script src="Scripts/wmtsqueries.js" type="text/javascript" ></script>
<script src="Scripts/wfsqueries.js" type="text/javascript" ></script>
<script src="Scripts/wmsqueries.js" type="text/javascript" ></script>
</head>
</html>
```

В откомпилированном виде инструментарий представлен библиотекой **gwtkse.js**.  
Подключение:

```
<script src=" gwtkse.js " type="text/javascript" ></script>
```

### 1.2.2 Системные требования

Поддерживается работа **GIS WebToolKit SE** в браузерах Microsoft Internet Explorer 8.0 и выше, Mozilla Firefox 3.6 и выше, Opera 11.0 и выше, Apple Safari 5.0 и выше и Chrome.

Рекомендуемые требования к техническим средствам:

- процессор: типа Intel Core i3 3.33 ГГц и выше;
- RAM – от 4Гб и выше.

## 2 ХАРАКТЕРИСТИКИ ПРОГРАММЫ

Классы и компоненты инструментария GIS WebToolKit SE предназначены для обмена данными с сервером с целью получения картографических данных и метаданных, выполнения специальных расчетов, оверлейных операций над объектами, выбора и поиска объектов.

Обмен данными выполняется через выполнение HTTP-запросов к сервису GIS WebService SE. GIS WebService SE поддерживает протоколы обмена пространственными данными по стандартам Open Geospatial Consortium (OGC): Web Map Tile Service (OGC WMTS) 1.1, Web Map Service (OGC WMS) 1.3.0, Web Feature Service (OGC WFS) 2.0, Web Coverage Service (OGC WCS) 2.0.1 и выполняет расширенные запросы с использованием REST API.

Классы инструментария используются для обмена данными с картографическим сервером без применения элементов интерфейса пользователя. Классы WmtsQueries, WmsQueries, WcsQueries, WfsQueries реализуют запросы данных по стандартным протоколам обмена OGC. Для выполнения специальных расчетов по карте используются дополнительные запросы.

Компоненты реализованы в виде классов, функций и элементов управления HTML. Компоненты Map, Layer позволяют встраивать интерактивную карту в Web-страницу.

В состав GIS WebToolKit SE входят:

- класс WmtsQueries – выполнение запросов по протоколу OGC WMTS;
- класс WmsQueries – выполнение запросов по протоколу WMS OGC;
- класс WcsQueries – выполнение запросов по протоколу WCS OGC;
- класс WfsQueries – выполнение запросов по протоколу WFS OGC;
- класс LayersControl – управление слоями карты;
- класс TileUtil – расчеты параметров тайлов;
- класс Util – анализ параметров и XML-данных, вспомогательные функции;
- класс LatLong – географическая точка;
- класс Point – точка на плоскости;
- класс handler – класс обработчиков событий карты;
- класс placemark – геообъект «Отметка на карте»;
- класс Projection – преобразования координат;
- компонент Map – карта;
- компонент Layer – слой тайлов карты;
- компонент WmsLayer – WMS-слой карты;
- компонент TextSearch – текстовый поиск в карте (по указанным атрибутам);
- компонент ruler – измерение расстояний по карте;
- компонент polygon – измерение площади произвольного многоугольника по карте.

### 3 ОБРАЩЕНИЕ К ПРОГРАММЕ

Классы WmtsQueries, WmsQueries, WfsQueries, WcsQueries обеспечивают получение картографических данных по протоколам OGC WMTS 1.1, OGC WMS 1.3.0, OGC WFS 2.0, OGC WCS 2.0.1. Методы перечисленных классов выполняют запросы данных и возвращают ответы сервера с использованием технологии AJAX. Запросы к сервису GIS WebService SE выполняются асинхронно.

Классы доступны для использования после загрузки в браузер файлов wqueries.js, wmtsqueries.js, wfsqueries.js, wmsqueries.js.

#### 3.1 Класс WmtsQueries

Класс WmtsQueries обеспечивает выполнение запросов к серверу для получения графических изображений карты в виде рисунков тайлов, получения метаданных объектов векторной карты и получения характеристик геоданных и параметров доступа к ним (метаданные карты).

Для использования класса необходимо загрузить в браузер файл **wmtsqueries.js**.

Запросы выполняются по протоколу OGC Web Map Tile Service (OGC WMTS) версии 1.0.0.

В качестве источника геоданных используется сервис карт GIS WebService SE. Методы класса **WmtsQueries** выполняют запросы GetCapabilities, GetTile и GetFeatureInfo сервиса.

Для создания объекта WmtsQueries может использоваться код:

```
var wmts = new WmtsQueries ("url");
```

*url* в общем виде может быть описан как «<http://Web-узел/виртуальный каталог/service.php>».

Например,

<http://gisserver.ru/GisWebserviceSE/service.php>

##### 3.1.1 Методы класса WmtsQueries

Метод	Описание
WmtsQueries ( url )	Конструктор. Выполняет инициализацию класса. url – адрес сервиса GIS WebService SE.
<i>Получение метаданных сервера</i>	
getcapabilities()	Выполнить запрос GetCapabilities. Возвращает метаданные в виде XML.
<i>Получение рисунков тайлов</i>	
tileurl( querytile )	Получить коллекцию строк запросов рисунков тайлов. Аргумент <i>querytile</i> – параметры запроса рисунков тайлов в кодировке JSON или объект JSON.  Функция возвращает коллекцию строк запросов тайлов (Url).  При ошибке входных параметров возвращает пустую коллекцию.

Метод	Описание
tileimage( <i>querytile</i> )	Получить изображение тайла. Аргумент <i>querytile</i> – параметры запроса рисунка тайла в кодировке JSON. Функция возвращает строку данных рисунка тайла.
<i>Получение метаданных о характеристиках объектов карты в точке</i>	
featureinfo ( <i>queryfeature</i> )	Получить метаданные объектов карты в точке. Аргумент <i>queryfeature</i> - параметры запроса информации об объекте. Функция возвращает строку метаданных объектов в запрошенном формате выдачи: HTML/XML или пустую строку при ошибке.

### 3.1.2 Обработчики событий при выполнении запросов к серверу

Запросы к серверу карт выполняются асинхронно с поддержкой междоменного интерфейса. Для получения данных от сервера необходимо назначить обработчик события *onload* запроса в классе WmtsQueries. Обработчик *onError* назначается автоматически, формирует xml-отчет об ошибке и передает его в качестве ответа.

Метод	Описание
onDataLoad ( <i>response</i> )	Обработчик события загрузки ответа сервера в браузер.
onError	Обработчик ошибок выполнения асинхронного запроса.

Ниже приводится пример кода для назначения обработчика **onDataLoad** класса WmtsQueries. Назначение обработчика необходимо для получения данных ответа сервера при асинхронном запросе. Событие *onload* происходит, когда данные от сервера получены в полном объеме.

```
function onDataLoaded ( response ) {
    alert( response);
    //Выполнение действий с полученными данными ...
}

...
var wmts = new WmtsQueries("http://gisserver.ru/GisWebserviceSE/service.php ");
wmts.onDataLoad = onDataLoaded;
```

### 3.1.3 Параметры запроса рисунков тайлов - аргумент *querytile*

В таблице представлено описание аргумента *querytile* методов класса WmtsQueries.

Аргумент *querytile* - JSON-строка или JSON-объект. Содержит значения параметров для выполнения запроса GetTile OGC WMTS.

Структура параметров:

Имя параметра	Описание
<i>layer</i>	Идентификатор слоя карты
<i>matrix</i>	Тип матрицы тайлов (GoogleMapsCompatible...)

style	Имя стиля слоя. Имя стиля по умолчанию-default
format	Формат рисунков тайлов карты. В соответствии со стандартом WMTS OGC возможные значения графических форматов вывода карты указываются в метаданных в элементе Layer/Format
zoom	Масштабный коэффициент отображения карты
min[col, row]	Наименьшие значения столбца и строки матрицы
max[col, row]	Наибольшие значения столбца и строки матрицы тайлов. При отсутствии этих параметров формируется ссылка для запроса одного тайла.

### 3.1.4 Параметры запроса информации об объекте - аргумент *queryfeature*

В таблице представлено описание аргумента *queryfeature* методов класса WmtsQueries.

Аргумент *queryfeature* - JSON-строка или JSON-объект. Содержит значения параметров для выполнения запросов GetFeatureInfo OGC WMTS.

Структура параметров:

Имя параметра	Описание
layer	Идентификатор слоя карты
tilematrixset	Тип матрицы тайлов (GoogleMapsCompatible...)
style	Имя стиля слоя. Имя стиля по умолчанию-default
format	Формат вывода рисунков тайлов карты. Возможные значения графических форматов вывода карты указываются в метаданных в элементе Layer/Format
tilematrix	Масштабный коэффициент отображения карты
tilerow, tilecol	Значения строки и столбца матрицы тайлов
info_format	Формат вывода информации (HTML/ XML)
feature_count	Наибольшее число объектов в ответе
i	Координата точки по горизонтали в изображении (в пикселях)
j	Координата точки по вертикали в изображении (в пикселях)
getid	Вывести в ответ идентификатор объекта (1/0)
area	Вывести в ответ значение площади объекта (1/0)
length	Вывести в ответ значение длины объекта (1/0)

## 3.2 Класс WmsQueries

Класс **WmsQueries** обеспечивает получение метаданных о доступных картах и возможностях сервера, получение изображений карты по известным географическим параметрам и получение метаданных объектов векторной карты.

Для использования класса необходимо загрузить в браузер файл **wmsqueries.js**.

Запросы выполняются по протоколу OGC Web Map Service (WMS OGC) версии 1.3.0.

В качестве источника геоданных используется сервис карт GIS WebService SE. Методы класса **WmsQueries** выполняют запросы GetCapabilities, GetMap и GetFeatureInfo сервиса.

Для создания объекта **WmsQueries** может использоваться код:

```
var wms = new WmsQueries ("url");
```

*url* в общем виде может быть описан как «<http://Web-узел/виртуальный каталог/service.php>». Например,

<http://gisserver.ru/GISWebserviceSE/service.php>

### 3.2.1 Методы класса WmsQueries

Метод	Описание
WmsQueries ( url )	Конструктор. Выполняет инициализацию класса. url – адрес сервиса GIS WebService SE.
<i>Получение метаданных сервера карт</i>	
getcapabilities()	Выполняет запрос GetCapabilities. Возвращает метаданные в виде XML.
<i>Получение ссылки для запросов рисунков карты</i>	
mapurl( querymap )	Создать строку запроса рисунка карты. Аргумент <i>querymap</i> – параметры запроса изображения карты. Функция возвращает строку запроса изображения карты (Url). При ошибке входных параметров возвращает пустую строку.
<i>Получение метаданных объектов карты в точке</i>	
featureinfo ( queryfeature )	Получить метаданные объектов карты в точке. Аргумент <i>queryfeature</i> - параметры запроса Getинформации об объекте. Функция возвращает строку метаданных объектов в запрошенном формате выдачи: HTML/ XML или пустую строку при ошибке.

### 3.2.2 Параметры запроса изображения карты - аргумент *querymap*

В таблице представлено описание аргумента *querymap* методов класса WmsQueries.

Аргумент *querymap* – JSON-строка или JSON –объект. Содержит значения параметров для выполнения запросов GetMap OGC WMST.

Структура параметров:

Имя параметра	Описание
layers	Список имен слоев (карт), разделенных запятой
styles	Список имен стилей слоев. Имя стиля по умолчанию-default .
crs	Имя референсной системы координат. Содержит числовой код проекции по EPSG.
bbox[]	Габариты области вывода карты в указанной CRS, две пары координат в виде «minx, miny, maxx, maxy». Указывают область картографирования на местности.
w	Ширина в пикселях рисунка карты
h	Высота в пикселях рисунка карты
format	Формат вывода рисунка карты. В соответствии со стандартом WMS OGC возможные значения графических форматов вывода карты указываются в метаданных в элементе Layer/Format

### 3.2.3 Параметры запроса информации об объекте – аргумент *queryfeature*

В таблице представлено описание аргумента *queryfeature* методов класса WmsQueries.

Аргумент *queryfeature* - JSON-строка или JSON-объект. Содержит значения параметров для выполнения запросов GetFeatureInfo OGC WMS.

Структура параметров:

Имя параметра	Описание
layers	Список имен слоев (карт), разделенных запятой
styles	Список имен стилей слоев. Имя стиля по умолчанию-default .
crs	Имя референсной системы координат. Содержит числовой код проекции по EPSG.
bbox[]	Габариты области вывода карты в указанной CRS, две пары координат в виде «minx, miny, maxx,maxy». Указывает область картографирования на местности.
w	Ширина в пикселях рисунка карты
h	Высота в пикселях рисунка карты
format	Формат вывода рисунков карты. В соответствии со стандартом WMS OGC возможные значения графических форматов вывода карты указываются в метаданных в элементе Layer/Format
layersquery	Список запрашиваемых имен слоев (карт), разделенных запятой
outputformat	Формат вывода информации (HTML/ XML)
maxcount	Наибольшее число объектов в ответе
i	Координата точки по горизонтали в изображении (в пикселях), $0 \leq i \leq w$
j	Координата точки по вертикали в изображении (в пикселях), $0 \leq j \leq h$

## 3.3 Класс WfsQueries

Класс **WfsQueries** обеспечивает получение картографической информации об объектах карты в виде набора векторной, атрибутивной и описательной информации, а также получение метаданных о доступных типах геопространственных объектов и возможностях сервера. Получение данных выполняется через выполнение HTTP-запросов по протоколу OGC Web Feature Service (OGC WFS и OGC WFS-T) версии 2.0 и расширенным запросам. Расширенные запросы выполняются через вызовы функций REST API сервиса карт.

Векторные данные объектов представляются в формате GML 3.2 (ISO 19136:2007).

В качестве источника геоданных используется сервис карт GIS WebService SE. Методы класса **WfsQueries** выполняют запросы GetCapabilities, DescribeFeatureType, GetFeature, ListStoredQueries, Transaction сервиса.

Входные данные для методов изменения объектов (транзакций) должны иметь кодировку XML/GML в соответствии с прикладной XSD-схемой.

Выходные данные методов транзакций содержат результат выполнения в виде XML.

Возможные значения и правила описания параметров приводятся в документе ПАРБ.00160-01 32 01 «Программное изделие GIS WebService SE Руководство системного программиста», раздел «Операции стандарта OGC WFS и OGC WFS-T».

Для использования класса необходимо загрузить в браузер файл **wfsqueries.js**.  
Для создания объекта **WfsQueries** может использоваться код:

```
var wfs = new WfsQueries ("url");
```

*url* в общем виде может быть описан как «<http://Web-узел/виртуальный каталог/service.php>». Например,

<http://gisserver.ru/GISWebserviceSE/service.php>

### 3.3.1 Методы класса WfsQueries

Метод	Описание
WfsQueries ( url )	Конструктор. Выполняет инициализацию класса. <i>url</i> – адрес сервиса GIS WebService SE.
<i>Получение метаданных сервера объектов</i>	
getfeaturetypenamelist ()	Выполняет запрос GetCapabilities. Возвращает метаданные в виде XML - список типов объектов, имеющихся в базе пространственных данных сервиса.
featuretype ( typenames )	Выполняет запрос DescribeFeatureType. Возвращает XSD-схему описания типов объектов в виде Xml. Описание типов объектов в схеме указывает, как экземпляры объектов будут кодироваться при выводе. Аргумент <i>typenames</i> содержит список типов объектов, при его отсутствии XSD-схема выводится полностью.
liststoredqueries ()	Получение информации о доступных хранимых запросах. Функция возвращает XML метаданных хранимых запросов.
<i>Получение векторных данных объектов</i>	
feature ( queryfeature )	Получить векторные и атрибутивные данные объектов карты. Аргумент <i>queryfeature</i> – параметры запроса GetFeature в JSON. Возвращает данные объектов в формате GML.
featurebyid( id )	Получить векторные и атрибутивные данные объекта по идентификатору. Аргумент <i>id</i> – уникальный идентификатор объекта карты. Возвращает данные одного объекта в формате GML, чей идентификатор совпал с указанным значением <i>id</i> .
<i>Изменение объектов - транзакции</i>	
transaction( action )	Выполнить транзакцию (изменение) набора данных. Аргумент <i>action</i> должен иметь XML формат команды wfs:Transaction/ wfs:Insert, wfs:Transaction/wfs:Replace или wfs:Transaction/wfs:Delete протокола OGC WFS для создания, замены или удаления объектов набора данных. Транзакции выполняются только в запросах в XML-кодировке методом POST. Возвращает XML-описание результата выполнения транзакции.

Метод	Описание
replace( replaceaction )	Заместить существующий экземпляр объекта набора данных. Аргумент <i>replaceaction</i> должен иметь XML формат команды wfs:Transaction/ wfs:Replace протокола OGC WFS. Возвращает XML-строку описания результата выполнения транзакции.
delete( deleteaction )	Удалить объект (объекты). Аргумент <i>deleteaction</i> должен иметь XML формат команды wfs:Transaction/ wfs:Delete протокола OGC WFS. Возвращает XML-строку описания результата выполнения транзакции
<i>Преобразование координат точек в указанную референсную систему координат</i>	
geopoint (layer, plane, crs )	Выполнить преобразование плоских прямоугольных координат точки в метрах в геодезические координаты в градусах по заданному коду EPSG. Аргумент <i>plane</i> – значения плоских прямоугольных координат точки в метрах в системе координат <i>crs</i> . Аргумент <i>crs</i> – код целевой проекции, строка. Имя референсной системы координат. Содержит префикс «EPSG» и числовой код проекции по классификации EPSG. Например «EPSG:4326». Аргумент <i>layer</i> – идентификатор слоя карты. Возвращает значения координат точки в градусах в указанной системе координат в формате GML, объект <i>gml:Point</i> .
planepoint(layer, geo, crs )	Выполнить преобразование геодезических координат точки в градусах в плоские прямоугольные координаты в метрах по заданному коду EPSG. Аргумент <i>geo</i> – значения геодезических координат точки в градусах в системе координат <i>crs</i> . Аргумент <i>crs</i> – код целевой проекции, строка. Имя референсной системы координат. Содержит префикс «EPSG» и числовой код проекции по классификации EPSG. Например «EPSG:4326». Аргумент <i>layer</i> – идентификатор слоя карты. Возвращает значения координат точки в указанной системе координат в формате GML, объект <i>gml:Point</i> .
<i>Поиск объектов по пространственному фильтру</i>	

Метод	Описание
areaseek(layer, xarea, method)	Выполнить поиск объектов, координаты которых пересекаются с заданной областью. Аргумент <i>layer</i> – идентификатор слоя карты для поиска. Аргумент <i>xarea</i> – описание области поиска в формате GML как FeatureCollection. Аргумент <i>method</i> – имя RESTметода поиска. Значение равно <i>areaseekcrossline</i> , если область поиска задана линией; <i>areaseekcrosssquare</i> , если область поиска задана полигоном. Функция возвращает список найденных объектов в формате GML как FeatureCollection.
areaseekinside(layer, xarea)	Выполнить поиск объектов, координаты которых полностью находятся внутри заданной области (вхождение в объект). Аргумент <i>layer</i> – идентификатор слоя карты для поиска. Аргумент <i>xarea</i> – описание области поиска в формате GML как FeatureCollection. Функция возвращает список найденных объектов в формате GML как FeatureCollection.
<i>Поиск объектов по текстовому фильтру</i>	
textsearch(layer, filter, index, count)	Выполнить текстовый поиск объектов по семантическим данным. Аргумент <i>layer</i> – идентификатор слоя карты для поиска. Аргумент <i>filter</i> – значение текстового фильтра в виде: ((список характеристик)(список условий сравнения)(список искомых значений)(логический оператор)). Список характеристик содержит имена характеристик ( свойств) объектов по xsd-схеме карты, например ObjName. Список условий содержит знаки сравнения «=,>,<,!=,>=,<=». Список значений содержит искомые текстовые значения семантических данных объектов. Каждый элемент списка значений указывается ключом «val=». Например, ( (ObjName, RoadNumber)(=,=)(val=Москва*val=*М-7*)(AND)). Логический оператор может принимать значение «OR» или «AND». При поиске по одной характеристике логический оператор не используется. Функция возвращает список найденных объектов в формате GML как FeatureCollection.
areafeature(layer, gmlpolygon)	Выполнить расчет площади полигона. Аргумент <i>layer</i> – идентификатор слоя карты. Аргумент <i>gmlpolygon</i> – описание полигона в формате GML.

### 3.3.2 Обработчики событий при выполнении запросов к серверу

Запросы к серверу карт выполняются асинхронно с поддержкой междоменного интерфейса. Для получения данных от сервера необходимо назначить обработчик onData-

Load события onload запроса в классе WfsQueries. Обработчик onError назначается автоматически, формирует xml-отчет об ошибке и передает его в качестве ответа в обработчик onDataLoad.

Метод	Описание
onDataLoad ( response )	Обработчик для получения данных от сервера.
onError	Обработчик ошибок выполнения асинхронного запроса.

Ниже приводится пример кода для назначения обработчика **onDataLoad** класса WfsQueries для получения данных ответа сервера при асинхронном запросе. Событие onload происходит, когда данные от сервера получены в полном объеме.

```
function onDataLoaded ( response ) {
    alert( response );
    //Выполнение действий с полученными данными ...
}

...
var wfs = new WfsQueries("http://gisserver.ru/GisWebserviceSE/service.php ");
wfs.onDataLoad = onDataLoaded;
```

### 3.3.3 Параметры запроса объекта - аргумент *queryfeature*

В таблице представлено описание аргумента *queryfeature* методов класса WfsQueries.

Аргумент *queryfeature* – JSON-строка или JSON-объект. Содержит значения параметров для выполнения запросов GetFeature OGC WFS.

Параметры имеют структуру:

Имя параметра	Описание
typenames	Список типов объектов через запятую, данные которых требуется получить. Стока.
srsname	Число, код проекции по EPSG. Параметр указывает, в какой системе координат сервер должен представить геометрию объектов в ответе.
bbox[]	Параметр указывает участок местности для отбора объектов. Массив, значения координат нижнего левого (LowerCorner) и верхнего правого (UpperCorner) углов прямоугольной области в виде последовательности координат и числового кода проекции по EPSG, в которой заданы координаты.
count	Число, ограничивает количество объектов в ответе сервера указанным значением.
startindex	Число, начальный номер объекта для помещения в документ ответа.
result	Число, способ вывода ответа операции. При result равном 1 ответ сервера содержит данные объектов в GML, при result равном 0 - XML, где содержится только общее количество объектов, соответствующих запросу. Значение параметра по умолчанию равно 1.

Возможные значения, правила описания параметров и результатов выполнения запросов приводятся в документе ПАРБ.00160-01 32 01 «Программное изделие GIS WebService SE Руководство системного программиста».

### 3.4 Класс WcsQueries

Класс **WcsQueries** обеспечивает получение пространственной информации о рельефе местности, а также получение описания условий получения геоданных и описания характеристик сервера по предоставлению этих данных. Получение данных выполняется через выполнение HTTP-запросов по протоколу OGC Web Coverage Service (OGC WCS) версии 2.0.1.

В качестве источника геоданных используется сервис карт GIS WebService SE. Методы класса **WcsQueries** выполняют запросы GetCapabilities, DescribeCoverage, GetCoverage сервиса GIS WebService SE.

Для использования класса необходимо загрузить в браузер файл **wquerieswcs.js**.

Для создания объекта **WcsQueries** может использоваться код:

```
var wcs = new WcsQueries ("url");
```

*url* в общем виде может быть описан как «<http://Web-узел/виртуальный каталог/service.php>». Например,

<http://gisserver.ru/GISWebserviceSE/service.php>

#### 3.4.1 Методы класса WcsQueries

Метод	Описание
WcsQueries (url)	Конструктор. Выполняет инициализацию класса. <i>url</i> – адрес сервиса GIS WebService SE.
<i>Получение метаданных сервера покрытий</i>	
getcapabilities()	Выполняет запрос GetCapabilities. Возвращает список трёхмерных растровых моделей местности в виде XML.
coveragelist(coverageid)	Получить полное описание покрытий (3D-моделей) по списку имен покрытий. Аргумент <i>coverageid</i> - строка, список покрытий, разделенных запятой. Возвращает метаданные полного описания по одной или нескольким моделям местности в формате XML с использованием описаний GML и SWE.
coverage(coverageoptions)	Получить данные о рельефе местности для указанного покрытия. Аргумент <i>coverageoptions</i> – параметры запроса GetCoverage в JSON. Возвращает данные о рельефе местности в формате XML с использованием GML и SWE.

#### 3.4.2 Параметры запроса поверхности - аргумент *coverageoptions*

В таблице представлено описание аргумента *coverageoptions* методов класса WcsQueries.

Аргумент имеет тип string в кодировке JSON. Содержит значения параметров для выполнения запросов GetCoverage OGC WCS.

Структура параметров:

Имя параметра	Описание
covarageid	Идентификатор поверхности
resolution	Разрешение поверхности(точность, шаг)
dimsubsetx	Ограничения для поверхности по оси X
dimsubsey	Ограничения для поверхности по оси Y

Возможные значения и правила описания параметров приводятся в документе ПАРБ.00160-01 32 01 «Программное изделие GIS WebService SE Руководство системного программиста».

### 3.5 Выполнение специальных расчетов по карте

Класс **MapMath** содержит набор методов для получения пространственных характеристик объектов карты, выполнения расчетов по карте. Для получения результатов выполняются расширенные запросы к сервису GIS WebService SE.

Класс доступен для использования после загрузки в браузер файла mapmath.js.

Для создания объекта **MapMath** может использоваться код:

```
var geomath = new MapMath ("url");
```

*url* в общем виде может быть описан как «<http://Web-узел/виртуальный каталог/service.php>». Например,

<http://gisserver.ru/GISWebserviceSE/service.php>

#### 3.5.1 Методы класса MapMath

Метод	Описание
MapMath ( url )	Конструктор. Выполняет инициализацию класса. <i>url</i> – адрес сервиса GIS WebService SE.
<i>Получение пространственных характеристик объектов карты</i>	
dimentions ( layer, id )	Получить длину, площадь, периметр объекта карты по идентификатору объекта. Аргумент <i>layer</i> – имя WMTS слоя карты. Аргумент <i>id</i> – уникальный идентификатор объекта карты. Возвращает длину, площадь, периметр одного объекта в формате GML, чей идентификатор совпал с указанным значением <i>id</i> . При ошибке возвращает XML-сообщение об ошибке в виде элемента ows:ExceptionReport.
dimentionfeatures (queryfeature)	Получить длину, площадь, периметр, семантические характеристики объектов карты. Аргумент <i>queryfeature</i> - параметры запроса информации об объектах, параметры «area» и «length» должны иметь значение «1».

area ( layer, id)	Получить площадь полигона. Аргумент <i>layer</i> – идентификатор слоя карты. Аргумент <i>id</i> – уникальный идентификатор объекта карты. Возвращает площадь и периметр полигона в GML. При ошибке возвращает XML-сообщение об ошибке в виде элемента ows:ExceptionReport.
areafeature(layer, gmlpolygon)	Получить площадь полигона. Аргумент <i>layer</i> – идентификатор слоя карты. Аргумент <i>gmlpolygon</i> – описание полигона в формате GML. Возвращает площадь полигона в GML. При ошибке возвращает XML-сообщение об ошибке в виде элемента ows:ExceptionReport.
length ( layer, id)	Получить длину линии. Аргумент <i>layer</i> – имя WMTS слоя карты. Аргумент <i>id</i> – уникальный идентификатор объекта карты. Возвращает длину линии в GML. При ошибке возвращает XML-сообщение об ошибке в виде элемента ows:ExceptionReport.
<i>Расчет азимута, дирекционного угла, расстояния по карте</i>	
azimuth ( layer, point1, point2, crs )	Получить значение азимута по координатам точек. Аргумент <i>layer</i> – имя WMTS слоя карты. Аргументы <i>point1</i> , <i>point2</i> – координаты точек в системе координат <i>crs</i> . Аргумент <i>crs</i> указывает EPSG код системы координат. Возвращает значение азимута в градусах в формате GML. При ошибке возвращает XML-сообщение об ошибке в виде элемента ows:ExceptionReport.
directionalangle ( layer, point1, point2, crs )	Получить значение дирекционного угла по координатам точек. Аргумент <i>layer</i> – имя WMTS слоя карты. Аргументы <i>point1</i> , <i>point2</i> – координаты точек в системе координат <i>crs</i> . Аргумент <i>crs</i> указывает EPSG код системы координат, в которой заданы координаты точек. Возвращает значение дирекционного угла в градусах в формате GML. При ошибке возвращает XML-сообщение об ошибке в виде элемента ows:ExceptionReport.
mapdistance ( layer, point1, point2, crs )	Получить значение геодезического расстояния по координатам точек. Аргумент <i>layer</i> – имя слоя карты. Аргументы <i>point1</i> , <i>point2</i> – координаты точек. Аргумент <i>crs</i> указывает EPSG код системы координат, в которой заданы координаты точек. Возвращает значение расстояния в метрах в формате GML. При ошибке возвращает XML-сообщение об ошибке в виде элемента ows:ExceptionReport.
<i>Обработка координат объектов</i>	

mappolygoncenter (layer, id )	Получить координаты центра площадного объекта. Аргумент <i>layer</i> – имя слоя карты. Аргумент <i>id</i> –идентификатор объекта карты gml:Id. Возвращает координаты центра в виде объекта gml:Point. При ошибке возвращает XML-сообщение об ошибке в виде элемента ows:ExceptionReport.
intersection( layer, feature, type )	Получить координаты пересечения линейного и площадного объектов. Аргумент <i>layer</i> – имя слоя карты. Аргумент <i>feature</i> –список идентификаторов (gml:Id) полигона и линии через запятую. Аргумент <i>type</i> – тип объектов результата расчета. Если <i>type</i> равно «0» выполняется расчет координат линейных объектов при пересечении полигона и линии. Если <i>type</i> равно «1» выполняется расчет координат площадных объектов при пересечении полигона и линии. Функция возвращает результат расчета пересечения объектов в формате GML как FeatureCollection. При ошибке возвращает XML-сообщение об ошибке в виде элемента ows:ExceptionReport.
intersectionsquare ( layer, feature )	Получить координаты площадных объектов при пересечении двух полигонов. Аргумент <i>layer</i> – имя слоя карты. Аргумент <i>feature</i> –список идентификаторов (gml:Id) полигонов через запятую. Функция возвращает результат расчета пересечения объектов в формате GML как FeatureCollection. При ошибке возвращает XML-сообщение об ошибке в виде элемента ows:ExceptionReport.
merging( layer, feature )	Получить координаты объединения объектов. Функция выполняет расчет координат площадных объектов при объединении двух полигонов. Аргумент <i>layer</i> - имя слоя карты. Аргументы <i>feature</i> - список идентификаторов (gml:Id) полигонов через запятую. Функция возвращает результат объединения объектов в формате GML как FeatureCollection. При ошибке возвращает XML-сообщение об ошибке в виде элемента ows:ExceptionReport.
spline( layer, feature, smoothing )	Построить метрику сплайна. Аргумент <i>layer</i> -имя слоя карты. Аргумент <i>feature</i> – векторные данные (координаты) объекта в формате GML, wfs:FeatureCollection/wfs:member/gml:LineString. Аргумент <i>smoothing</i> – строка, тип сплайна: сглаживающий или огибающий. При <i>smoothing</i> равном «smooth», строится сглаживающий сплайн, при «bend» – огибающий. Функция возвращает координаты сплайна в формате GML- gml: LineString.

### 3.5.2 Ошибки выполнения расчетов

Сообщения сервиса GIS WebService SE об ошибках при выполнении расчетов выводятся в виде XML по стандарту OGC OWS 2.0.

Сообщение об ошибке представляет собой элемент ows:ExceptionReport:

```
<?xml version="1.0" encoding="utf-8"?>
<ExceptionReport version="1.0.0" xmlns="http://www.opengis.net/ows/2.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    schemaLocation="http://www.opengis.net/ows/2.0 owsExceptionReport.xsd">
    <Exception code="" locator="">
        <ExceptionText></ExceptionText>
    </Exception>
</ExceptionReport>
```

## 3.6 Компонент Карта Map

Компонент Карта содержит набор классов, функций и элементов управления HTML для создания интерактивной карты на Web-странице.

При инициализации компонента карты происходит динамическое формирование HTML-кода и внедрение его в DOM-дерево страницы, где размещена карта.

Карта состоит из одного или нескольких наложенных друг на друга слоёв изображений. Слой может представлять различный тип картографической информации – космические снимки, векторные карты, тематические данные. Каждый слой разбит на множество квадратных участков — тайлов. При отображении карты тайлы совмещаются, слои накладываются друг на друга и формируется единое изображение. Формирование единого изображения производится автоматически, этот процесс скрыт от пользователя.

Map - основной класс инструментария **GIS WebToolKit SE**, используется для создания, удаления объекта карты и управления картой на Web-форме.

Основными параметрами карты являются область отображения карты, картографические параметры и параметры настройки карты.

Область отображения карты определяется HTML-элементом для размещения рисунка карты. Параметры определяют место картографирования и выполняемые операции на карте.

Для создания объекта Map может использоваться функция вида:

```
var map = null;
function init(options)
{
    map = new Map (divid, options);
}
```

Для удаления карты используется метод **Map.destroy()**.

Конструктор Map принимает два параметра: идентификатор HTML-элемента карты и параметры описания карты.

Карта может быть размещена в любом блочном HTML-элементе и полностью заполняет занимаемую им прямоугольную область. Обычно используется элемент div.

Изображение карты имеет многослойную структуру, каждый слой может представлять различные типы данных карты – снимки, векторные карты, схемы, 3D-модели местности.

Карта содержит коллекцию слоев данных. Класс Layer используется для управления отдельным слоем карты.

Параметры карты, заданные при инициализации, могут быть изменены в дальнейшем.

### 3.6.1 Параметры карты

Параметры карты содержат данные для инициализации класса карты и классов слоев тайлов карты.

Параметры описываются в кодировке JSON и имеют структуру:

Имя параметра	Описание
id	Уникальный идентификатор карты
url	Адрес сервера
center	Геокоординаты центра карты , [B,L]
tilematrix	Уровень масштабирования карты
minzoom	Предел уменьшения изображения карты, уровень масштабирования
maxzoom	Предел увеличения изображения карты, уровень масштабирования
crs	EPSG код проекции карты, число
tilematrixset	Тип матрицы тайлов (GoogleMapsCompatible...)
count	Число слоев карты
layers[]	Параметры слоев карты, массив. Структура, элемента массива описана в разделе «Параметры слоя карты»
mapcontenttree	Адрес файла настройки дерева состава карты

Например,

```

<head>
<script type="text/javascript">

var options = {
    "url": "http://gisserver.info/GISWebServiceSE/service.php", "id": "200",
    "center": [55.50, 38.39],
    "tilematrix": 10,
    "crs": 3857,
    "tilematrixset": "GoogleMapsCompatible",
    "count": 1,
    "layers": [{"id": "Noginsk", "alias": "Ногинский район", "url": "http://gisserver.info/GISWebServiceSE/service.php?Service=WMTS&REQUEST=GetTile&VERSION=1.0.0&LAYER=0002 &TILEMATRIXSET=%tilematrixset&TILEMATRIX=%z&TILEROW=%y&TILECOL=%x&FORMAT=image%2Fpng&&STYLE=default" }],
    "mapcontenttree": "configmapcontent.xml"
};

map = Map("divMap", options);

</script>
</head>
<body>
    <p>Моя Карта</p>
    <div id="divMap" style="width:800px; height:600px"></div>
</body>
```

### 3.6.2 Методы управления состоянием карты

Для выбора отображаемой области и масштаба отображения карты используются следующие методы.

Метод	Описание
setView ( centerpoint, z)	Установить отображение карты (географический центр отображаемого фрагмента и уровень масштабирования карты)
setViewport ( centerpoint)	Установить положение карты и отобразить
setZoom (zoom)	Установить уровень масштабирования карты
setBounds(bounds, crs)	Установить габариты карты. Ограничить размер карты в указанных пределах значений.
zoomIn(delta)	Увеличить изображение карты на delta шагов
zoomOut(delta)	Уменьшить изображение карты на delta шагов
destroy()	Удалить карту и очистить обработчики событий карты
showMap()	Отобразить тайлы карты (все видимые слои)
setMatrix	Установить тип матрицы тайлов
hideLayer(id)	Отключить отображение слоя карты
showLayer(id)	Включить отображение слоя карты

### 3.6.3 Методы получения информации о карте

Метод	Описание
getSize()	Возвращает размер карты
getCenter ()	Возвращает географический центр карты
getZoom ()	Возвращает масштаб отображения карты
getBounds()	Возвращает габариты карты
getMaxZoom()	Возвращает знаменатель масштаба максимально-го увеличения изображения карты
getMinZoom()	Возвращает знаменатель масштаба минимально-го уменьшения изображения карты
getLayersCount()	Возвращает число слоев карты
getMapTopLeft()	Возвращает координаты верхнего левого угла отображаемого фрагмента карты
getZoomScale(zoom)	Возвращает значение масштабного коэффициен-та для указанного уровня масштабирования кар-ты
getWindowSize()	Возвращает ширину и высоту окна карты

### 3.6.4 Методы управления слоями карты

Метод	Описание
addLayer(layerparam)	Добавить слой карты.
removeLayer(id)	Удалить слой карты.
hideLayer(id)	Скрыть отображение слоя по идентификатору
showLayer(id)	Отобразить слой

resetMap(options, remove)	Обновить карту. <i>options</i> - параметры карты (пункт 3.6.1), <i>options.layers</i> - параметры добавляемых слоев карты (новых). <i>remove</i> - список идентификаторов слоев, которые необходимо закрыть. При <i>remove</i> = '*' удаляются все текущие слои карты.
showLayersExclusively(layerid)	Отобразить только указанные слои карты (исключивно). <i>layerid</i> – список идентификаторов слоев через запятую.

### 3.6.5 Методы назначения обработчиков событий

Метод	Описание
initEvents()	Назначить обработчики событий мыши

### 3.6.6 Методы отбора объектов карты

Метод	Описание
getFeatureInfo(point)	Запросить информацию об объектах карты в точке (i,j) окна карты (пиксели)
onDataLoadedWmts(response, context)	Функция обратного вызова для операции выбора объектов карты в точке

### 3.6.7 Методы настройки интерфейса пользователя

Метод	Описание
initPanes()	Инициализация панелей интерфейса карты
setMapContents()	Настроить панель «Состав карты»
setTextSearch()	Настроить компоненты текстового поиска

### 3.6.8 Методы управления оверлеями карты

Оверлеи карты - это геообъекты, которые добавляются поверх всех слоев карты. Объекты-оверлеи привязаны к географическим координатам и перемещаются при перетаскивании или масштабировании карты. К оверлеям относятся метки карты (placemark). Метка карты описывается классом **placemark**. Метки карты хранятся в карте в массиве *placemarks*. Визуальные элементы меток располагаются в панели оверлеев карты.

Следующие методы используются для создания, удаления, обновления объектов-оверлеев карты.

Метод	Описание
overlayAppend(geo, point, view, id)	Добавить метку в массив и на карту
overlayClear()	Удалить метки на карте (в панели оверлеев)
overlayRefresh()	Обновить метки на карте (панель оверлеев)
placemarkRemove()	Удалить метки на карте и очистить список меток

### 3.6.9 События класса

Компонент инициализирует события для HTML-элемента карты.

Событие	Описание
mapclick	<p>Клик манипулятором мышь по карте</p> <p>Параметры:</p> <ul style="list-style-type: none"> <li>point – координаты точки в единицах экрана,</li> <li>coord – координаты точки в метрах в проекции карты,</li> <li>geo – геодезические координаты точки</li> </ul> <p>Пример использования:</p> <pre>\$('#divMap').on('mapclick', function (ui) {     // alert(ui.point.x + ', ' + ui.point.y); });</pre>
featurelistclick	<p>Выбор объекта в окне найденных объектов карты</p> <p>Параметры:</p> <ul style="list-style-type: none"> <li>layer – id слоя карты,</li> <li>gid – идентификатор объекта</li> </ul> <p>Пример использования:</p> <pre>\$('#divMap').on('featurelistclick', function (ui) {     // alert(ui.layer + ', ' + ui.gid); });</pre>
overlayRefresh	<p>Обновление (перерисовка) карты</p> <p>Пример использования:</p> <pre>\$('#divMap').on('overlayRefresh', function (ui) {     //alert('overlayRefresh'); });</pre>

### 3.7 Компонент Слой карты Layer

Класс Layer обеспечивает загрузку, отображение и управление отдельным слоем тайловой карты. Основным параметром слоя является адрес сервера тайлов (url).

Слои создаются при создании карты, хранятся в коллекции **layers** класса карты Map. Конструктор Layer принимает два параметра – ссылку на объект карты Map и описание параметров слоя. После создания объекта Layer необходимо добавить его в коллекцию слоев карты методом onAdd().

```
var lay = new Layer(map, options);
lay.onAdd();
```

#### 3.7.1 Параметры слоя карты

Параметры слоя служат для инициализации слоя карты. Описываются в кодировке JSON и имеют структуру:

Имя	Описание
id	Уникальный идентификатор слоя тайлов
alias	Название слоя
url	Шаблон запросов изображений данных (тайлов)
format	Формат рисунков тайлов карты (png/jpg)
style	Имя стиля карты. default по умолчанию.
matrixSet[]	Список поддерживаемых матриц тайлов

Имя	Описание
visible	Признак видимости слоя (1/0)
selectObject	Возможность выбора объектов карты (1/0)
legend	Параметр легенды карты. Список типов объектов, условное обозначение которых отображается в дереве состава карты. Если значение списка равно «*», отображаются условные обозначения всех типов объектов.
keyssearchbyname	Параметры фильтра объектов по названию (текстовый фильтр). Список ключей характеристик вида [key1,key2,...,keyN]
iconUrl	Адрес пиктограммы слоя
minZoomView	Граница видимости слоя минимальная
maxZoomView	Граница видимости слоя максимальная
sheet	Имя листа из паспорта карты

### 3.7.2 Методы класса Layer

Метод	Описание
setOptions()	Установить параметры слоя
init(map, options)	Инициализация класса
onAdd()	Добавить слой в карту
onRemove()	Удалить слой
show()	Отобразить слой
hide()	Скрыть отображение слоя
getMinZoomView()	Запросить масштабный коэффициент минимальной границы видимости слоя
getMaxZoomView()	Запросить масштабный коэффициент максимальной границы видимости слоя
getVisibility()	Запросить видимость слоя
isTextSearch()	Запросить возможность текстового поиска объектов в слое
firstDraw()	Создать матрицу тайлов слоя
update()	Обновить слой карты
setShift(dx, dy)	Сдвинуть матрицу тайлов слоя на (dx, dy) пиксел

### 3.7.3 Подключение внешних геопорталов в качестве слоев карты

В качестве источников пространственных данных могут использоваться внешние геопорталы. Для получения данных геопортала необходимо создать слой карты, указать адрес сервера и описать параметры запроса.

Например, для подключения геопортала Росреестра в параметрах карты необходимо описать слой:

```
{
  "id": "RR", "alias": "Кадастровые данные Росреестра", "url": "http://maps.rosreestr.ru/arcgis/services/Cadastre/CadastreWMS/MapServer/WMS?REQUEST=GetMap&SERVICE=WMS&VERSION=1.3.0&LAYERS=6,7,8,9,10,11,12,18,19,20,21,22,23&STYLES=&FORMAT=image%2Fpng&BGCOLOR=0xFEFEFE&HEIGHT=%h&WIDTH=%w&TRANSPARENT=TRUE&CRS=EPSG:3857&BBOX=%bbox"
}
```

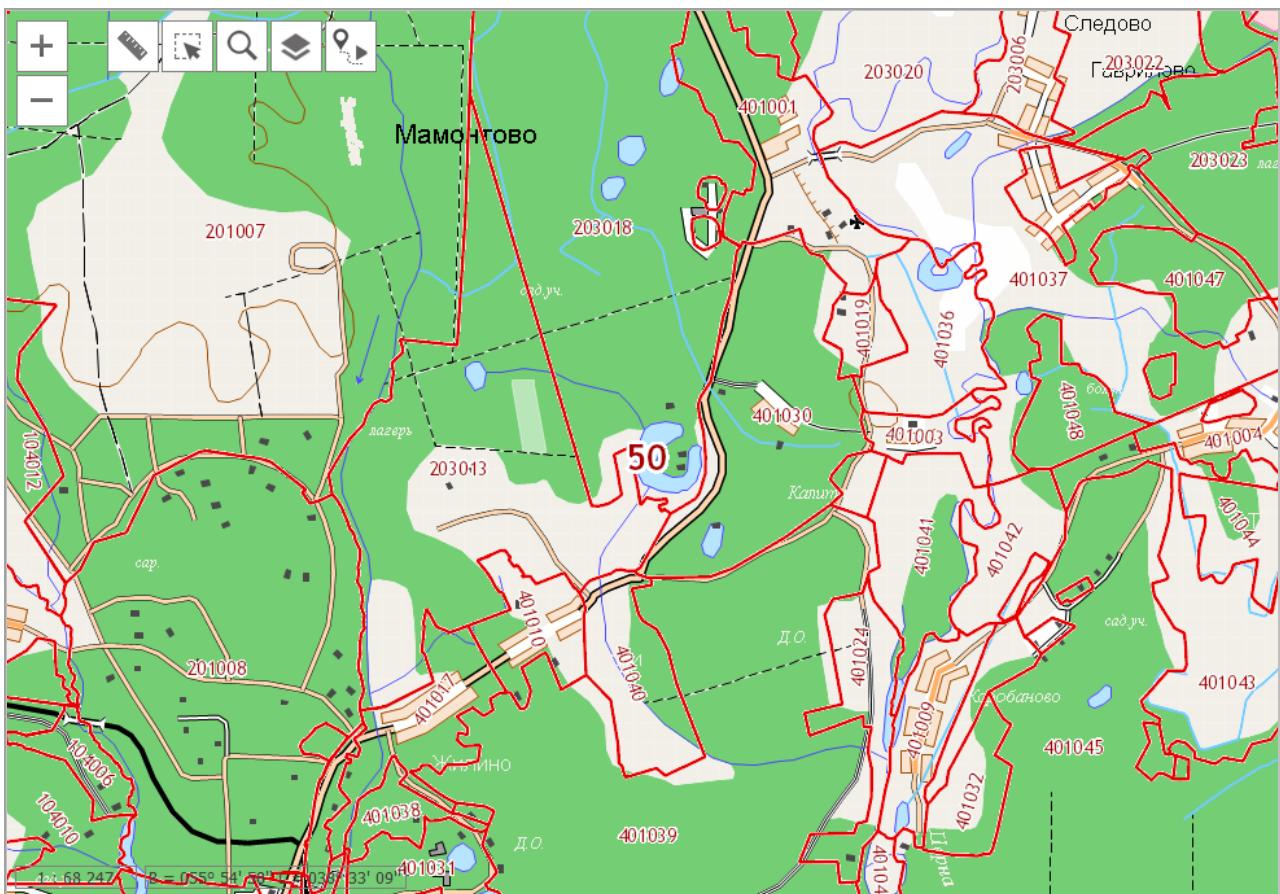


Рис.1 Отображение Кадастровой карты Портала услуг Росреестра

### 3.8 Компонент WMS-слой карты WmsLayer

Класс WmsLayer обеспечивает загрузку, отображение и управление отдельным WMS-слоем карты. Основным параметром слоя является адрес картографического сервера (url).

Слои создаются при создании карты, хранятся в коллекции **layers** класса карты Map. Конструктор WmsLayer принимает два параметра – объект Map и описание параметров слоя. Параметры слоя служат для инициализации WMS-слоя карты, их структура описана в пункте 3.7.1 настоящего документа.

После создания объекта Layer необходимо добавить его в коллекцию слоев карты методом onAdd().

```
var lay = new GWTK.WmsLayer(map, options);
lay.onAdd();
```

#### 3.8.1 Методы класса WmsLayer

Метод	Описание
setOptions()	Установить параметры слоя
init(map, options)	Инициализация класса
onAdd()	Добавить слой в карту
onRemove()	Удалить слой
show()	Отобразить слой
hide()	Скрыть отображение слоя
getVisibility()	Запросить видимость слоя

update()	Обновить слой карты
buildGetMap	Запросить ссылку запроса GetMap слоя

### 3.9 Компонент Состав карты MapContentControl

Компонент **MapContentControl** предназначен для отображения и управления составом видимых слоев карты через пользовательский интерфейс.

Компонент отображает список слоев карты в виде иерархического дерева данных.

Для управления видимостью слоя карты необходимо включить/выключить кнопку соответствующего элемента дерева.

Для описания структуры и вида дерева слоев используется JSON объект, который указывается в параметрах карты через параметр - «*contenttree*».

Структура JSON объекта приводится в разделе 0 (Описание структуры дерева компонента **MapContentControl**).

Компонент реализован с использованием виджета w2sidebar открытой библиотеки w2ui.

Для использования **MapContentControl** необходимо подключить библиотеку скриптов w2ui\w2ui-1.4.2.min.js и файл стилей w2ui\w2ui-1.4.2.min.css.

```
<link href=" w2ui/w2ui-1.4.2.css" rel="stylesheet" />
<script src=" w2ui/w2ui-1.4.2.min.js"></script>
```

Компонент **MapContentControl** программно связан с картой, инициализируется в классе карты.

#### 3.9.1 Методы класса MapContentControl

Метод	Описание
initMapContentPane( <i>parent</i> )	Инициализация компонента, где <i>parent</i> - ссылка на панель, в которой отображается компонент.
createTree()	Создать пустое дерево.
destroy()	Деинициализация компонента.

addItems([parentId], ItemData)	<p>Добавить элемент дерева, где <i>parentId</i> - уникальный идентификатор родительского узла дерева (необязательный); <i>ItemData</i>- описание узла в дереве.</p> <p>Параметры <i>ItemData</i>:</p> <ul style="list-style-type: none"> <li><i>ItemId</i> - уникальный идентификатор узла дерева (для управлением слоем <i>ItemId</i> должен быть равен <i>layers[]</i>.xId, а <i>group</i> = false);</li> <li><i>alias</i> - текстовое название узла (имя слоя, имя папки или имя группы);</li> <li><i>expanded</i> - состояние группы (узла): развернута (true), свернута (false);</li> <li><i>group</i> - признак группы (папки): false - узел предназначен для управления слоем, иначе узел представляет собой группу (при <i>parentId</i> = null) или папку (при <i>parentId</i> = &lt;идентификатор родительского узла&gt;);</li> <li><i>img</i> – имя стиля css, для отображения пиктограммы узла дерева;</li> <li><i>checked</i> - состояние флагка для управлением слоем;</li> </ul> <p>При успешном добавлении слоя, функция возвращает значение <i>layerData.ItemId</i>.</p>
delItems(ItemId)	Удалить элементы дерева по идентификатору <i>ItemId</i> .
clearAllItems ()	Удалить все элементы дерева.
setChecked(ItemId)	Установить флаг для слоя по идентификатору <i>ItemId</i> .
setUnChecked(ItemId)	Снять флаг слоя по идентификатору <i>ItemId</i> .
readcfgfile(xmlcfgTreeFile)	Заполнить дерево из файла настройки, где <i>xmlcfgTreeFile</i> ссылка на файл описания структуры дерева.
ItemChecked(event)	Событие при установке/снятии флагка управления слоем. При снятии флагка отключает видимость слоя в карте.
ReplaceItemText(ItemId)	Обновление содержания HTML кода узла дерева. Необходимо для корректного отображения состояния флагков после выполнения операций свернуть/развернуть узле.
ItemHide(ItemId)	Скрыть узел.
ItemShow(ItemId)	Показать узел.
readjsonstr(json_str)	Добавить узлы из объекта JSON
prepareItemText()	Выполнить подготовку HTML текста узла дерева. Необходимо вызывает до добавления узлов. Данный метод к текстовому описанию добавляет картинку и/или флагок управления слоем.

### 3.9.2 Описание структуры дерева компонента MapContentControl

Настройки дерева компонента MapContentControl осуществляется с помощью JSON объекта. JSON объект представляет собой массив узлов дерева в иерархическом виде.

Описание узла дерева:

Наименование	Описание	Значение по умолчанию
id	Уникальный идентификатор узла дерева	null
text	Текстовое содержание узла дерева (может содержать HTML код)	"
clickable	Признак необходимости добавления флагка управления слоем	true
t_img	Ссылка на изображение	"
nodes	Массив подчиненных узлов дерева	[]
expanded	Развернуть подчиненные узлы дерева	false
hidden	Скрыть узел дерева	false
group	Признак группы	false

Пример:

```
var contenttree = [
    { id: 'Worldmap', text: 'Карта мира', t_img:'images\content.png', clickable: true },
    { id: 'RR', text: 'Кадастровое деление, земельные участки, объекты капитального строительства', t_img:'images\content.png', expanded: true, clickable: true,
        nodes: [
            {id: 'RR_1', text: 'Кадастровые округа'},
            {id: 'RR_2', text: 'Кадастровые районы'},
            {id: 'RR_3', text: 'Кадастровые кварталы'},
            {id: 'RR_4', text: 'Объекты недвижимости'}
        ]
    }
];
```

## 3.10 Компонент текстового поиска объектов карты mapTextSearch

Компонент **mapTextSearch** предназначен для выполнения запросов поиска пространственных объектов карты по текстовому фильтру, получения ответа на запросы и отображения результатов поиска.

Доступ к пространственным данным объектов карты выполняется через выполнение расширенных запросов с использованием REST API WFS к серверу карт. Компонент **mapTextSearch** получает ответы сервера в формате XML/GML и отображает результат поиска с использованием DOM объектов браузера и элементов управления HTML.

Текстовый поиск в карте выполняется по семантическим данным (характеристикам) объектов карты. Для слоя карты необходимо установить список ключей характеристик, используемых при поиске. Список указывается в параметрах слоя, имя параметра – *keyssearchbyname*, раздел 3.7.1 данного документа. Если параметры слоя не содержат список *keyssearchbyname*, поиск объектов в таком слое не выполняется.

Компонент **mapTextSearch** программно связан с картой, инициализируется в классе карты.

Конструктор **mapTextSearch** принимает один параметр – ссылку на объект Map.

```
...
map.mapTextSearch = new GWTK.mapTextSearch (map);
```

Подключаемый файл - textsearch.js:

```
<script src="source/core/searchobjects.js"></script>
```

### 3.10.1 Методы класса mapTextSearch

Метод	Описание
setTextSearchPane()	Настроить панель текстового поиска
setTextSearchPagination (index,maxcount)	Настроить панель навигации текстового поиска
textSearchStarting(text, index)	Выполнить текстовый поиск объектов в карте по шаблону text. index – начальный индекс записи в ответе сервера.
textSearchNext()	Продолжить поиск вперед.
textSearchPrev()	Продолжить поиск назад.
refreshResult()	Обновить панель объектов карты (результат поиска)
onDataLoadedWfs(response, context)	Обработчик ответа операции текстового поиска в карте. Выводит результат поиска на панель объектов карты в виде HTML.
getfeaturecollectionHtml(xml)	Получить HTML из XML/GML результата поиска.

## 3.11 Компонент Панель поиска TextSearchControl

Компонент **TextSearchControl** обеспечивает интерфейс пользователя при поиске пространственных объектов карты по текстовому фильтру. Компонент инициализируется в классе карты, если указано в параметрах карты (раздел 3.7.1 и 3.9 данного документа). Если ни один из слоев карты не содержит параметры поиска объектов, панель поиска не отображается.

Конструктор **TextSearchControl** принимает один параметр – ссылку на панель для размещения элементов управления (div).

```
...
var tscnt = new GWTK.TextSearchControl(this.textSearchPane);
```

Подключаемый файл - textsearchcontrol.js:

```
<script src="source/ui/textsearchcontrol.js "></script>
```

### 3.11.1 Методы класса TextSearchControl

Метод	Описание
initTextSearchPane()	Настроить панель поиска объектов
textValue()	Запросить значение строки поиска. Функция возвращает значение строки поля ввода.

## 3.12 Компонент метка placemark

Компонент **placemark** реализует геообъект типа точка (метка).

Экземпляр карты размещает метки в собственном хранилище в виде массива объектов, ссылка на который находится в поле *placemarks*. Добавление метки на карту, ее изменение и удаление производится с помощью обращения к этому массиву при помощи методов класса карты.

Пример добавления метки в карту:

```
var geo = GWTK.toLatLng([55.86, 38.349]);  
  
var overlaypoint = GWTK.tileView.geo2pixelOffset(map, geo);  
  
var id = 'mark100';  
  
var show = true;  
  
map.overlayAppend(geo, overlaypoint, show, id);
```

Пример кода для создания геообъекта метки:

```
// создать метку в точке geo  
var mark = new GWTK.placemark(geo);  
  
// добавить в массив меток карты  
map.placemarks.push(mark);  
  
// установить положение в окне карты (точка привязки)  
mark.position(point);  
  
// отобразить в панели оверлеев  
map.overlayPane.appendChild(mark.geopoint());
```

Конструктор **placemark** (latlng, title, text, imgurl) принимает четыре параметра:

- latlng – географические координаты точки;
- title – текст подсказки метки;
- text - текст метки;
- imgurl – Url изображения метки.

### 3.12.1 Методы класса placemark

Метод	Описание
create()	Создать визуальный объект метки
position(point)	Установить точку привязки метки в окне, point – прямоугольные координаты точки в пикселях в окне карты
geopoint()	Возвращает визуальный объект метки (div)
toPoint()	Возвращает координаты метки (B,L) в виде GWTK.point

## 3.13 Компонент Редактор карты mapeditor

Элемент редактирования слоев карты.

Компонент mapeditor предназначен для создания и редактирования картографической информации.

Компонент реализован с использованием виджета w2sidebar открытой библиотеки w2ui.

Для использования rosreestr необходимо подключить библиотеку скриптов w2ui\w2ui-1.4.2.min.js и файл стилей w2ui\w2ui-1.4.2.min.css.

```
<link href="w2ui/w2ui-1.4.2.css" rel="stylesheet" />
<script src="w2ui/w2ui-1.4.2.min.js"></script>
```

Конструктор mapedit (id, map, param) принимает три параметра:

- id – уникальный идентификатор объекта;
- map – ссылка на объект Map
- param – параметры инициализации объекта в кодировке JSON, описание параметров дано в пункте 3.15.1 (Описание параметров инициализации объекта mapeditor);

### 3.13.1 Описание параметров инициализации объекта mapeditor

Имя параметра	Описание
maplayersid	<p>Редактируемые слои</p> <p>Функциональные возможности редактора:</p> <ul style="list-style-type: none"> <li>- пустой массив, если поддерживаются все операции,</li> <li>- массив заданных режимов, если есть ограничения:</li> </ul> <p>"create" - создание,      "edit" - редактирование,      "delete" - удаление,      "group" - групповые операции</p>

### 3.13.2 Пример использования компонента mapeditor

Код визуализации:

```
<script>
var options = {
  "url": "http://gisserver.info/GISWebServiceSE/service.php", "id": "200", "center": [55.86, 38.349], "tilematrix": 13, "crs": 3857, "tilematrixset": "GoogleMapsCompatible", "maxzoom": 17, "minzoom": 0,
  "layers": [
    {
      "id": "NoginskEditing", "alias": "Ногинск редактирование", "url": "http://gisserver.info/GISWebServiceSE/service.php?SERVICE=WMS&REQUEST=GetMap&VERSION=1.3.0&FORMAT=image%2Fpng&LAYERS=NoginskEdit&BBOX=%bbox&HEIGHT=%h&WIDTH=%w&CRS=%crs&dt=%dt",
      "selectObject": 1, "keyssearchbyname": ["ObjName"], "selectsearch": 1
    }
  ],
  "errorTileUrl": "http://gisserver.ru/gwtkse/images/empty.gif", "controls": ["ruler", "textsearchinmap"], "url_addresssearch": ["http://geocode-maps.yandex.ru/1.x/?geocode=", "Искать в Яндекс"]
};
```

```
var options_mapEditor =
{
    "maplayersid": ["NoginskEditing"] // редактируемые слои данных
    , "functions": [] // функциональные возможности ("create", "edit", "delete",
"group")
};

// Инициализация карты
function initMap() {
    if (window.GWTK) {
        theMap = new GWTK.Map("dvMap", options);

        if (theMap != null && theMap != undefined)
            var mapEditor = new GWTK.mapEditor('1', theMap, options_mapEditor);
    }
}

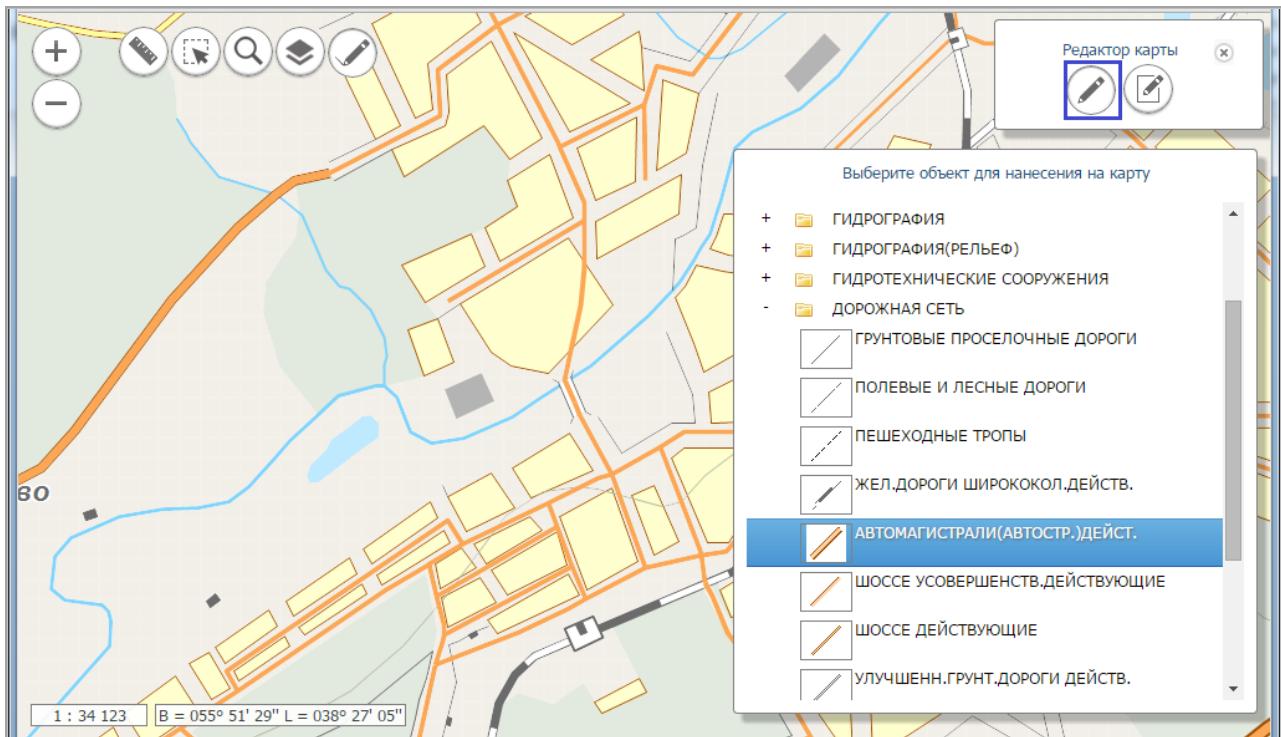
</script>

<body onload="initMap(); " style="padding:0px; margin:0px; border:0px;">
    <div id="dvMap" style="width:100%; height:100%;">
    </div>
</body>
```

Результат старта редактора карты:



Режим создания объекта карты, выбор объекта



### 3.14 Компонент polygon

Компонент **polygon** предназначен для выбора произвольного многоугольника на карте, расчета его площади и отображения результата. Выбор выполняется путем указания точек вершин многоугольника кликом мыши в изображении карты. Для расчета площади выполняется асинхронный запрос к картографическому сервису GIS WebService SE.

Экземпляр класса **polygon** размещается в классе карты, имя поля – **polygonArea**.

Конструктор **polygon (map)** принимает один параметр,  
где **map** – ссылка на объект карты **Map**.

Например,

```
var map = new GWTK.Map( divid, options );
...
map.polygonArea = new GWTK.polygon(map);
```

Для удаления экземпляра компонента необходимо вызвать метод **destroy**:

```
map.polygonArea.destroy();
map.polygonArea = null;
```

#### 3.14.1 Методы класса polygon

Метод	Описание
<b>addPolygonPoint (event)</b>	Добавить вершину многоугольника
<b>setPolygonHint</b>	Установить окно вывода результата
<b>refresh()</b>	Обновить изображение многоугольника
<b>getArea()</b>	Выполнить запрос площади
<b>areaResponse(response, context)</b>	Вывести результат расчета площади
<b>closeLine</b>	Замкнуть контур многоугольника
<b>removeLastSegment()</b>	Разомкнуть контур
<b>destroy()</b>	Удалить данные объекта

### 3.15 Компонент sliderToolsControl

Комбинированный элемент управления слайдером.

Компонент sliderToolsControl предназначен для изменения значения бегунка слайдера при помощи кнопок управления, работает с данными типа Date.

Компонент реализован с использованием открытой библиотеки jQuery UI: горизонтального слайдера (Slider Widget) и кнопки (Button Widget).

Для использования sliderToolsControl необходимо подключить библиотеки скриптов jquery.maskedinput.js, jquery-ui-1.11.2.min.js и файл стилей jquery-ui.css.

```
<link href="jquery-ui-1.11.2/jquery-ui.css" rel="stylesheet" />
<script src="plugins/jquery-ui-1.11.2.min.js"></script>
<script src="plugins/jquery.maskedinput.js"></script>
```

Основными параметрами компонента являются идентификатор слайдера, область размещения кнопок управления, область размещения бегунка слайдера, параметры инициализации. Области определяются HTML-элементами. Обычно используется элемент div.

Конструктор sliderToolsControl(id, toolsdiv, sliderdiv, initparam) принимает четыре параметра:

- id – уникальный идентификатор объекта;
- toolsdiv – область размещения кнопок управления (HTML DOM Object);
- sliderdiv – область размещения бегунка слайдера (HTML DOM Object);
- initparam – параметры инициализации – структурированный массив из одного элемента, где
  - initparam.dtbegin - время начала в формате Date()
  - initparam.dtend - время окончания начала в формате Date(),
  - initparam.step - шаг в миллисекундах,
  - initparam.interval - интервал обновления панели слайдера в режиме проигрывания в миллисекундах,
  - initparam.mininterval - минимально допустимый интервал.

#### 3.15.1 Методы класса sliderToolsControl

Метод	Описание
initSlider(dtbegin, dtend, step, interval, mininterval)	<p>Инициализация слайдера, вызывается при отсутствии параметра initparam в конструкторе.</p> <ul style="list-style-type: none"><li>• dtbegin - время начала в формате Date()</li><li>• dtend - время окончания начала в формате Date(),</li><li>• step - шаг в миллисекундах,</li><li>• interval - интервал обновления панели слайдера в режиме проигрывания в миллисекундах,</li><li>• mininterval - минимально допустимый интервал.</li></ul>

<code>setValue (value)</code>	Установка текущего значения бегунка слайдера, где value - значение в миллисекундах
<code>getValue ()</code>	Запрос текущего значение бегунка слайдера
<code>getSlider()</code>	Визуальный объект ползунка слайдера (div)
<code>getSliderWidget()</code>	Объект Slider Widget библиотеки jQuery UI, используется для доступа к методам и событиям объекта Slider Widget
<code>getButton(type)</code>	Визуальный объект кнопки, где type – тип кнопки по назначению: 'beginning' - в начало 'end' - в конец 'forward' - шаг вперед 'rewind' - шаг назад 'play' - проигрывание вперед 'playback'- проигрывание назад 'wrench' - параметры
<code>getButtonWidget(type)</code>	Объект Slider Button библиотеки jQuery UI, используется для доступа к методам и событиям объекта Slider Button, где type – тип кнопки по назначению: 'beginning' - в начало 'end' - в конец 'forward' - шаг вперед 'rewind' - шаг назад 'play' - проигрывание вперед 'playback'- проигрывание назад 'wrench' - параметры
<code>stop()</code>	Остановка запущенного слайдера
<code>destroy ()</code>	Удаление объекта sliderToolsControl

### 3.15.2 События класса sliderToolsControl

Компонент инициализирует события для HTML-элемента области размещения бегунка слайдера (в конструкторе параметр sliderdiv). Помимо этих событий пользователь может обрабатывать события, которые инициализируют плагины, используемые компонентом sliderToolsControl.

Событие	Описание

<p><code>updatedataSliderToolsControl</code></p>	<p>Изменение значения бегунка слайдера.</p> <p>Параметры:</p> <ul style="list-style-type: none"> <li>- <code>dt</code> – значение в миллисекундах,</li> <li>- <code>service</code> - направление движения:</li> <ul style="list-style-type: none"> <li>- "begin" - начало</li> <li>- "end" - конец</li> <li>- "step_forward" - шаг вперед</li> <li>- "step_back" - шаг назад</li> <li>- "random" - случайно выбранное значение</li> <li>- "start" - старт</li> <li>- "stop" - остановка</li> <li>- "playprocess" – проигрывание вперед</li> <li>- "playprocess_back" – проигрывание назад</li> </ul> </ul> <p>Пример использования:</p> <pre>\$('#sliderPanel').on('updatedataSliderToolsControl',   function (ui) {     // alert(ui.service);   }); }</pre>
<p><code>updateparamSliderToolsControl</code></p>	<p>Изменение значений в диалоге параметров слайдера.</p> <p>Параметры:</p> <ul style="list-style-type: none"> <li>- <code>step</code> – шаг в миллисекундах,</li> <li>- <code>interval</code> – интервал обновления панели слайдера в режиме проигрывания в миллисекундах</li> </ul> <p>Пример использования:</p> <pre>\$('#sliderPanel').on('updateparamSliderToolsControl',   function (ui) {     // alert(ui.step);   }); }</pre>

### 3.15.3 Пример использования компонента sliderToolsControl

Код визуализации:

```
<body>





```

или

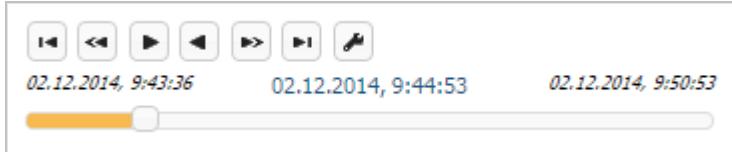
```

<body>
    <table width="350px">
        <tr> <td> <div id="ToolTips"></div> </td> </tr>
        <tr> <td> <div id="sliderPanel" > </div> </td> </tr>
    </table>
</body>

<script>
$(document).ready(function () {
    var sliderRoute = new GWTK.sliderToolsControl("slider2", $("#ToolTips")[0], $("#sliderPanel")[0]);
    sliderRoute.initSlider(new Date(141750261600), new Date(1417503053000), 250, 250, 250);
});
</script>

```

Результат выполнения.



### 3.16 Компонент Маршрутизация и анализ данных с БПЛА routeBPLA.

Комбинированный элемент визуализации маршрута с синхронной демонстрацией видеофайла.

Компонент routeBPLA предназначен для создания маршрута по данным файла KML (OpenGIS® KML Encoding Standard) или csv, содержащих временные метки, с синхронным просмотром отснятого видеоматериала по данным, созданных беспилотным летательным аппаратом (БПЛА). Просмотр видеофайлов возможен только в браузерах, поддерживающих стандарт HTML5.

Компонент реализован с использованием открытой библиотеки jQuery UI.

Для использования routeBPLA необходимо подключить библиотеки скриптов jquery.maskedinput.js, jquery-ui-1.11.2.min.js и файл стилей jquery-ui.css.

```

<link href="jquery-ui-1.11.2/jquery-ui.css" rel="stylesheet" />
<script src="plugins/jquery-ui-1.11.2.min.js"></script>
<script src="plugins/jquery.maskedinput.js"></script>

```

Конструктор routeBPLA(id, param, mapdiv, sliderdiv, videodiv) принимает пять параметров:

- id – уникальный идентификатор объекта;
- param – параметры инициализации объекта в кодировке JSON, описание параметров дано в пункте 3.12.1 (Описание параметров инициализации объекта routeBPLA);
- mapdiv – область размещения точек-маркеров маршрута, панель объекта Map (HTML DOM Object);
- sliderdiv – область размещения слайдера (HTML DOM Object), при отсутствии размещается в mapdiv;
- videodiv – область размещения видеоплеера (HTML DOM Object), при отсутствии размещается в mapdiv.

### 3.16.1 Описание параметров инициализации объекта routeBPLA

Имя параметра	Описание
alias	Название объекта
file	Ссылка на файл kml или csv
checkpoint	Признак отображения всех точек созданного маршрута (0 или 1)
videovisible	Признак отображения видеоплеера при запуске (0 или 1)
videospeed	Скорость воспроизведения видео (0,1,2)
currmovi	Номер текущего набора видеоматериала (начиная с 1)
movies [ {"file": http://62.173.139.13/files/BPLA.MP4 " "timebegin": 0 }]	Массив структур видеоданных, где file - ссылка на видеофайл в формате mp4 timebegin - сдвиг относительно начала трека координат в миллисекундах
fn_pointTooltip(number, points[number-1])	Альтернативная функция вывода подсказки для текущей точки (параметры: номер точки с 1, элемент массива описания точки points). Функция должна вернуть строку подсказки
fn_geo2pixelOffset(BL[])	Альтернативная функция пересчета координат точки в координаты окна карты (параметр: массив координат BL[2] или XY[2]). Функция должна вернуть объект GWTK.Point)

### 3.16.2 Методы класса routeBPLA

Метод	Описание
initdataFromfile(currentpoint, videoVisible, videoIsPlay, currentmovies)	Инициализация данных из файла, где - currentpoint – текущая точка (с 1), - videoVisible – признак отображения видеоплеера - videoIsPlay – признак отображения видеоплеера в режиме проигрывания - currentmovies – текущий набор видеоданных (с 1)

pointsdata()	<p>Запрос массива структур points - основных данных класса.</p> <p>Массив содержит описание временных меток, используемых для построения маршрута. Массив автоматически инициализируется данными файла kml. При загрузке данных из csv-файла массив инициализируется функцией пользовательской fn_initpointscsv([[]]).</p> <p>Обязательными полями элемента массива points являются:</p> <ul style="list-style-type: none"><li>В - широта (тип float),</li><li>L - долгота (тип float),</li><li>H – высота (тип int)</li><li>time_millisecond - смещение в миллисекундах относительно начала трека (тип int),</li><li>datetime_utc – дата (тип Date),</li><li>realtime – реальная дата в миллисекундах, вычисляется автоматически(тип float).</li></ul> <p>Дополнительными полями являются:</p> <ul style="list-style-type: none"><li>ascent_feet</li><li>speed_mph</li><li>distance_feet</li><li>max_altitude_feet</li><li>max_ascent_feet</li><li>max_speed_mph</li><li>max_distance_feet</li><li>datetime_local</li><li>satellites</li><li>pressure_Pa</li><li>temperature_F</li></ul> <p>В дальнейшем пользователь может обрабатывать данные этих полей по своему усмотрению во время обработки событий, инициализируемых компонентом.</p> <p>Количество дополнительных полей может быть любым, но следует учитывать, что только с указанным набором работают функции savedata () и getdata().</p>
--------------	--

initdataFromdata (points, currentpoint, videoVisible, videoIsPlay, currentmovies)	Инициализация данных из массива структур points, где - points – массив структур points. Обязательными полями элемента массива points являются: В - широта (тип float), L - долгота (тип float), time_millisecond - смещение в миллисекундах относительно начала трека (тип int), datetime_utc – дата (тип Date) - currentpoint – текущая точка (с 1), - videoVisible – признак отображения видеоплеера - videoIsPlay – признак отображения видеоплеера в режиме проигрывания - currentmovies – текущий набор видеоданных (с 1)
refresh ()	Обновление данных в части отображения картографической информации после изменения параметров отображения карты (масштабирование, сдвиг и пр.)
savedata ()	Сохранение массива структур points в локальном хранилище данных localStorage (пункт ‘createBPLA_routepoints’)
getdata()	Восстановление данных из локального хранилища localStorage (пункт ‘createBPLA_routepoints’) в массив структур points
destroy()	Удаление объекта routeBPLA

### 3.16.3 События класса routeBPLA

Компонент инициализирует события для HTML-элемента области размещения точек-маркеров маршрута (в конструкторе параметр mapdiv).

Событие	Описание
filecompleteBPLA	Полная загрузка файла исходных данных (kml или csv)  Пример использования: \$(mapPane).on('filecompleteBPLA', function (ui) { // alert(ui); });
fileerrorBPLA	Ошибка при загрузке файла исходных данных (kml или csv)  Пример использования: \$(mapPane).on('fileerrorBPLA ', function (ui) { // alert(ui.error); });

updatedatarouteBPLA	<p>Изменение текущей точки.</p> <p>Параметры:</p> <p style="margin-left: 20px;">point – элемент массива points, pointnumber – номер точки с 1:</p> <p>Пример использования: \$(mapPane).on('updatedatarouteBPLA ', function (ui) { // alert(ui.pointnumber); });</p>
updateparamBPLA	<p>Обновление значений параметров диалога слайдера.</p> <p>Параметры:</p> <p style="margin-left: 20px;">step – шаг в миллисекундах, interval – интервал обновления панели слайдера в режиме проигрывания в миллисекундах</p> <p>Пример использования: \$(mapPane).on('updateparamBPLA ', function (ui) { // alert(ui.step); });</p>
videovisibleBPLA	<p>Событие на включение/отключение видимости окна видеоплеера</p> <p>Пример использования: \$(mapPane).on('videovisibleBPLA ', function (ui) { // alert(ui.value); });</p>
videoplayBPLA	<p>Событие на запуск/останов видеоплеера</p> <p>Пример использования: \$(mapPane).on('videoplayBPLA ', function (ui) { // alert(ui.value); });</p>
checkpointBPLA	<p>Изменение флага отображения всех точек объекта BPLA</p> <p>Пример использования: \$(mapPane).on('checkpointBPLA ', function (ui) { // alert(ui.value );});</p>
videospeedBPLA	<p>Изменение скорости проигрывания видеоплеера</p> <p>Пример использования: \$(mapPane).on('videospeedBPLA ', function (ui) { // alert(ui.value);});</p>

#### 3.16.4 Пример использования компонента routeBPLA.

Поскольку временные метки располагаются в окне карты, при создании компонента наличие компонента Карта (Map) обязательно.

Код визуализации:

```
<script>

var options = {
    "url": "http://gisserver.info/GISWebServiceSE/service.php", "id": "200",
    "center": [55.86, 38.349], "tilematrix": 13, "crs": 3857,
```

```

"tilematrixset": "GoogleMapsCompatible", "count": 2, "maxzoom": 23, "minzoom": 0,
"layers": [
{
  "id": "OSM", "alias": "Россия",
  "url": "http://gisserver.info/GISWebServiceSE/service.php?SERVICE=WMTS&REQUEST=GetTile&VERSION=1.0.0&LAYER=Osm&STYLE=default&TILEMATRIXSET=%tilematrixset&TILEMATRIX=%z&TILEROW=%y&TILECOL=%x&FORMAT=image%2Fpng"
}}
}

// Инициализация карты
function initMap() {
  if (window.GWTK) {
    theMap = new GWTK.Map("dvMap", options);
    if (theMap != null && theMap != undefined)
      testrouteBPLA(theMap.mapPane);
  }
}

// Запуск маршрутизатора
function testrouteBPLA(mapPane) {
  $(document).ready(function () {
    var route = {
      "alias": "test_kml",
      "file": "http://62.173.139.13/files/BPLA.kml",
      "checkpoint": 1,
      "videovisible": 1,
      "movies": [
        { "file": "http://62.173.139.13/files/BPLA.MP4", "timebegin": 0 }
      ],
      "fn_showCenter": showCenter
    }

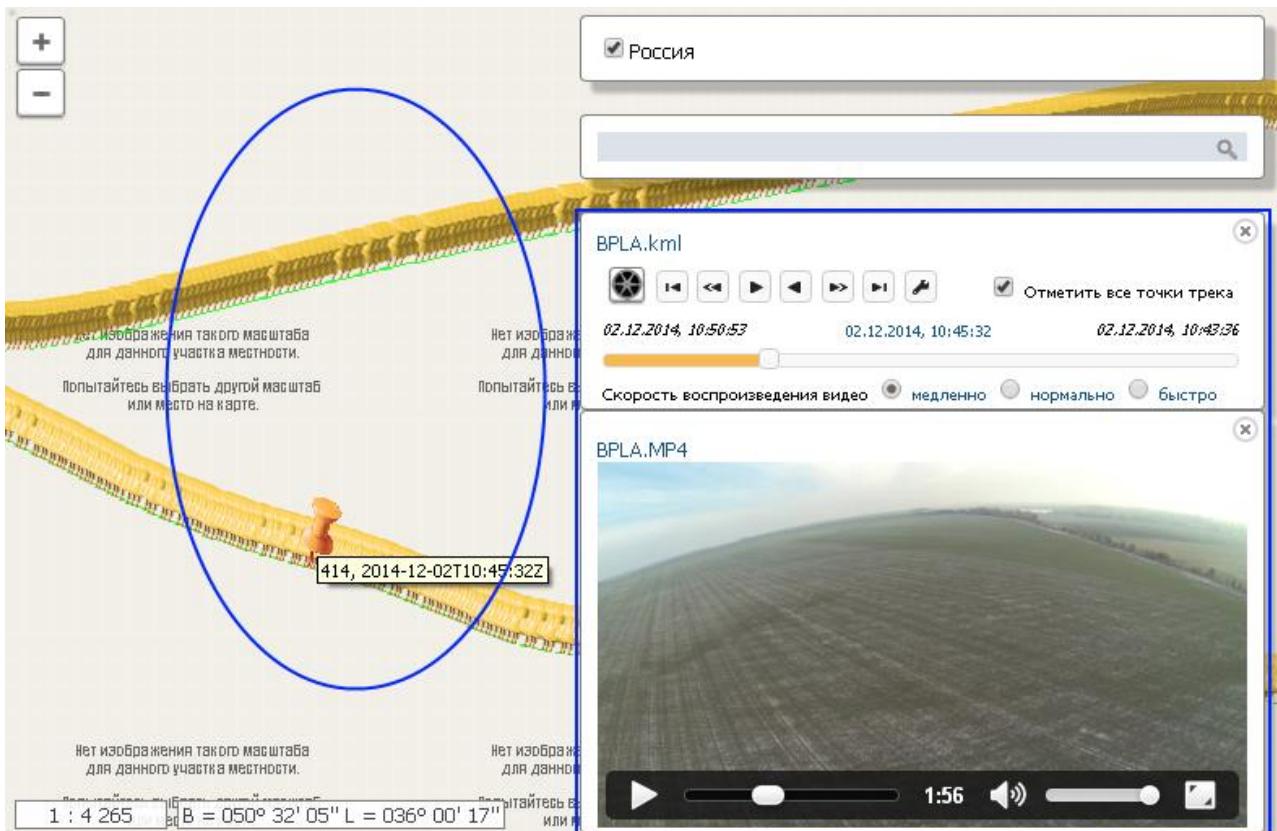
    var maproute = new GWTK.routeBPLA("BPLA", route, mapPane);
    $(mapPane).on('filecompleteBPLA', function (ui) {
      if (timerRouteld != 0) {
        clearTimeout(timerRouteld);
        timerRouteld = 0;
      }
    });
    timerRouteld = setTimeout(function () {
      maproute.initdataFromfile();
    }, 1000);
  });
}

// Переместить карту в центр маршрута
function showCenter (centerObj)
{
  var layer = 0;
  if (centerObj == null || centerObj.length == 0) return;
  var tilematrix = null;
  if (GWTK.maphandlers.map.layers[layer].options.tilematrix != null)
    tilematrix = GWTK.maphandlers.map.layers[layer].options.tilematrix;
  GWTK.maphandlers.map.setView(null, centerObj, tilematrix);
}
</script>

<body onload="initMap();>
<div id="dvMap" style="width:100%; height:100%;>
</div>
</body>

```

### Результат выполнения



### 3.17 Компонент Отображение сведений государственного кадастра недвижимости rosreestr

Элемент отображения сведений государственного кадастра недвижимости.

Компонент rosreestr предназначен для запроса и отображения информации государственного кадастра недвижимости для заданной точки на карте.

Компонент реализован с использованием виджета w2tabs открытой библиотеки w2ui.

Для использования rosreestr необходимо подключить библиотеку скриптов w2ui\w2ui-1.4.2.min.js и файл стилей w2ui\w2ui-1.4.2.min.css.

```
<link href="w2ui/w2ui-1.4.2.css" rel="stylesheet" />
<script src="w2ui/w2ui-1.4.2.min.js"></script>
```

Конструктор компонента принимает один параметр – ссылка на объект Map.

#### 3.17.1 Пример использования компонента rosreestr

Код визуализации:

```
<script>
var options = {
    "url": "http://gisserver.info/GISWebServiceSE/service.php", "id": "200",
    "center": [55.86, 38.349], "tilematrix": 13, "crs": 3857,
    "tilematrixset": "GoogleMapsCompatible", "count": 2, "maxzoom": 23, "minzoom": 0,
    "layers": [
        {
    "id": "Noginsk", "alias": "Ногинский район", "bbox": [55.5909, 37.865, 56.1799, 38.885], "selectObject": 1,
    "keyssearchbyname": ["ObjName"], "selectsearch": 1,
```

```

"url": "SER-
VICE=WMTS&REQUEST=GetTile&VERSION=1.0.0&LAYER=0001&STYLE=default&TILEMATRIXSET=%t
ilematrixset&TILEMATRIX=%z&TILEROW=%y&TILECOL=%x&FORMAT=image%2Fpng"    }]}
}

// Инициализация карты
function initMap() {
    if (window.GWTK) {
        theMap = new GWTK.Map("dvMap", options);

        if (theMap != null && theMap != undefined)
            var rosreestr = new GWTK.rosreestr(theMap);

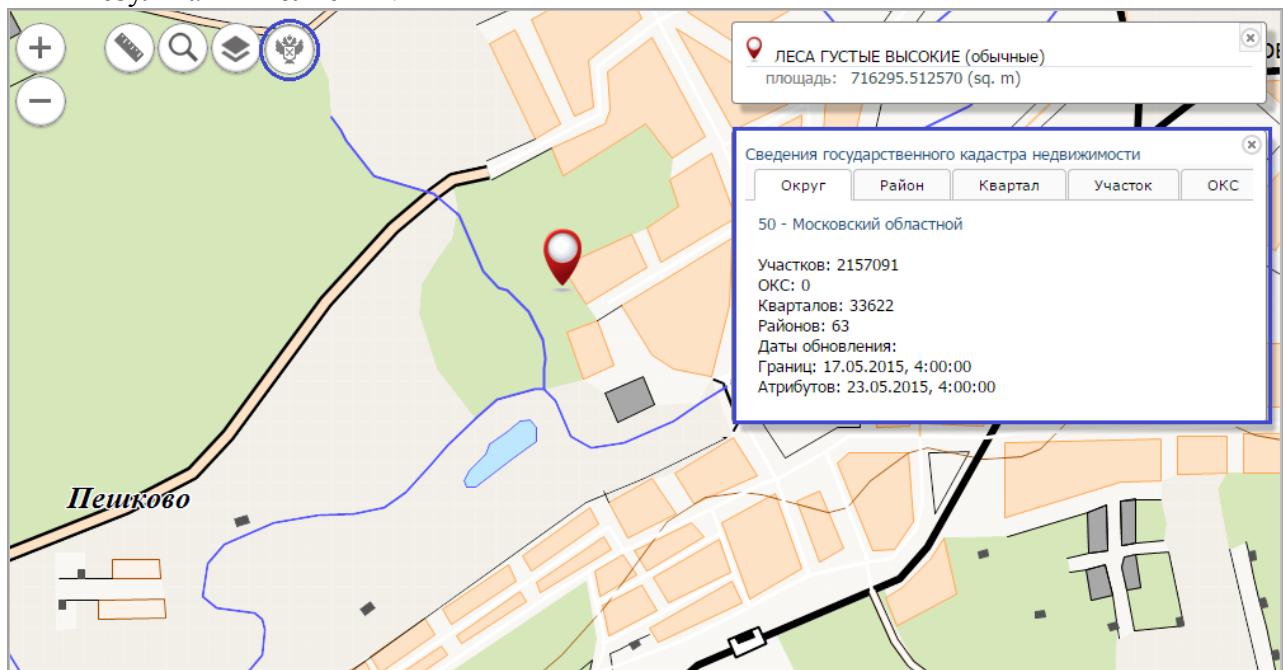
    }
}

</script>

<body onload="initMap();" style="padding:0px; margin:0px; border:0px;">
    <div id="dvMap" style="width:100%; height:100%;">
    </div>
</body>

```

Результат выполнения:



Компонент rosreestr включает в себя элемент rosreestrControl, который может использоваться самостоятельно. Описание компонента rosreestrControl дано в пункте 3.14.2.

### 3.17.2 Компонент rosreestrControl

Элемент отображения сведений государственного кадастра недвижимости.

Компонент rosreestrControl предназначен для запроса и отображения информации государственного кадастра недвижимости для заданной точки на карте.

Компонент реализован с использованием виджета w2tabs открытой библиотеки w2ui.

Для использования rosreestrControl необходимо подключить библиотеку скриптов w2ui\w2ui-1.4.2.min.js и файл стилей w2ui\w2ui-1.4.2.min.css.

```
<link href="w2ui/w2ui-1.4.2.css" rel="stylesheet" />
<script src="w2ui/w2ui-1.4.2.min.js"></script>
```

Компонент работает самостоятельно или входит в состав объекта rosreestr.

Конструктор rosreestrControl (id, parentdiv, mapdiv, x, y, activetab) принимает параметры:

- id – уникальный идентификатор объекта;
- parentdiv – область размещения компонента ((HTML DOM Object));
- mapdiv – панель объекта Map (HTML DOM Object);  
(не является обязательным, если parentdiv - панель объекта Map);
- x, y – координаты точки для запроса данных (в метрах, код EPSG:3857, Меркатора на шаре, принятом в Google), необязательный параметр;
- active – тип текущей закладки при запуске компонента, необязательный параметр, где: 0 - объекты капитального строительства, 1 - участки, 2 - кварталы, 3 - районы, 4 – округа.

Инициализация компонента данными государственного кадастра недвижимости происходит при задании координат точки в конструкторе компонента или после выбора точки на карте.

### 3.17.2.1 Методы класса rosreestrControl

Метод	Описание
destroy()	Удаление объекта rosreestrControl
show(show)	Отобразить/скрыть компонент Параметр show принимает значения: true – отобразить, false - скрыть
setPoint(x,y)	Установить текущую точку. Параметры x, y – координаты точки в метрах в системе EPSG:3857
onclick(active)	Считать данные активной закладки. Параметр active принимает значения: 0 - объекты капитального строительства, 1 - участки, 2 - кварталы, 3 - районы, 4 – округа.

### 3.17.2.2 События класса rosreestrControl

Компонент инициализирует события для HTML-элемента области размещения.

Событие	Описание
changetabRosreestr	Изменение текущей закладки. Параметры: active – номер активной закладки (от 0 до 4) Пример использования: \$(mapPane).on('changetabRosreestr', function (ui) { // alert(ui.active);});

### 3.17.2.3 Пример использования компонента rosreestrControl.

Код визуализации:

```

<script>
var options = {
  "url": "http://gisserver.info/GISWebServiceSE/service.php", "id": "200",
  "center": [55.86, 38.349], "tilematrix": 13, "crs": 3857,
  "tilematrixset": "GoogleMapsCompatible", "count": 2, "maxzoom": 23, "minzoom": 0,
  "layers": [
    {
      "id": "Noginsk", "alias": "Ногинский район", "bbox": [55.5909, 37.865, 56.1799, 38.885], "selectObject": 1,
      "keyssearchbyname": ["ObjName"], "selectsearch": 1,
      "url": "SER-
      VICE=WMTS&REQUEST=GetTile&VERSION=1.0.0&LAYER=0001&STYLE=default&TILEMATRIXSET=%t
      ilematrixset&TILEMATRIX=%z&TILEROW=%y&TILECOL=%x&FORMAT=image%2Fpng"      }]}
}

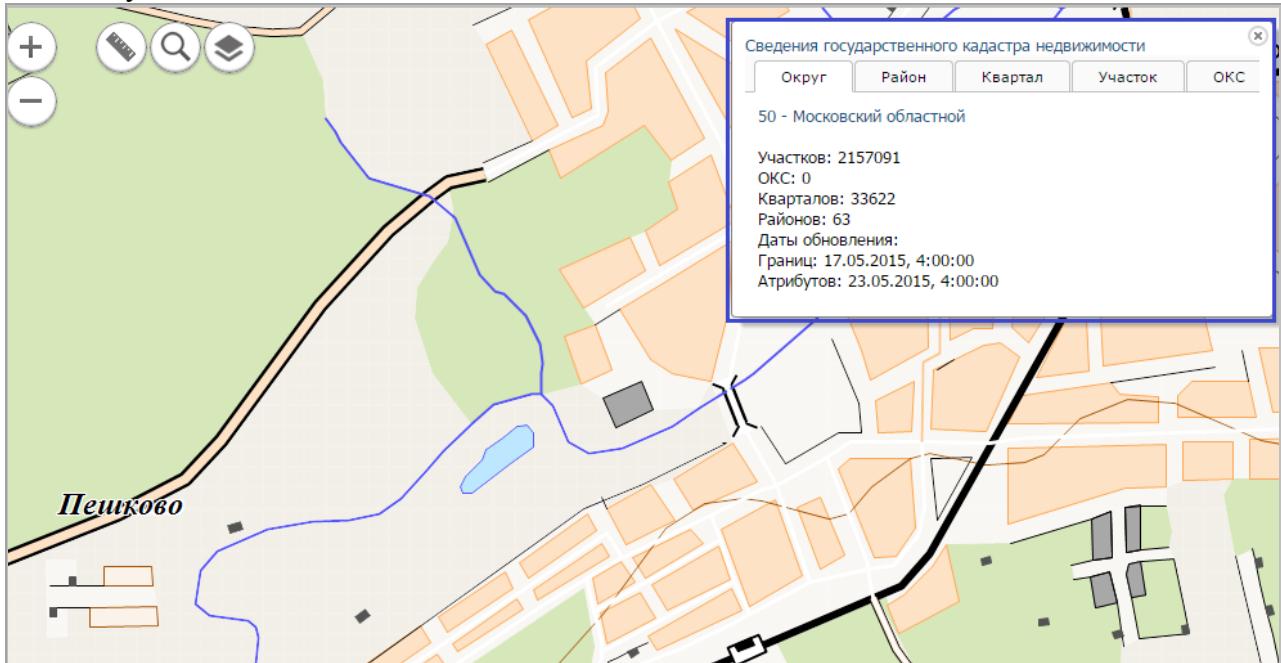
// Инициализация карты
function initMap() {
  if (window.GWTK) {
    theMap = new GWTK.Map("dvMap", options);
    if (theMap != null && theMap != undefined)
      var rosreestr = new GWTK.rosreestrControl('1', theMap.mapPane);
  }
}

</script>

<body onload="initMap();" style="padding:0px; margin:0px; border:0px;">
  <div id="dvMap" style="width:100%; height:100%;">
  </div>
</body>

```

Результат выполнения:



## **4 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ**

Входные данные программы размещаются на Web-сервере. Доступ к данным выполняется через выполнение запросов по протоколу HTTP. Могут использоваться различные типы карты: данные ДЗЗ, векторные, матричные карты. Источниками пространственных данных выступают картографические Web-серверы.

Получение растровых изображений карты выполняется по протоколу OGC WMTS, OGC WMS или другому, обеспечивающему выдачу тайлов в виде рисунков карты в формате графических файлов png и jpg.

Получение информации об объектах карты выполняется в формате XML/HTML через выполнение запросов GetFeatureInfo по протоколу OGC WMTS, OGC WMS.

Получение векторных данных объектов (геометрии), атрибутивной и описательной информации объектов карты выполняется по протоколу OGC WFS, данные объектов представляются в формате GML 3.2 (ISO 19136:2007).

Получение информации о матрицах высот выполняется по протоколу OGC WCS.

## 5 СООБЩЕНИЯ ПРОГРАММЫ

### 5.1 Сообщения класса WmtsQueries

При возникновении ошибок выполнения запросов к серверу выполняется обработчик onError. Обработчик назначается автоматически, формирует XML-сообщение об ошибке «Request Error» и передает его в качестве ответа.

Отчет формируется по стандарту OGC OWS 2.0 и имеет вид:

```
<?xml version="1.0"?>
<ExceptionReport version="1.0.0" xmlns="http://www.opengis.net/ows/2.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/ows/2.0 owsExceptionReport.xsd" >
    <Exception>
        <ExceptionText>Request Error </ExceptionText>
    </Exception>
</ExceptionReport>
```

## 6 ПРОВЕРКА ПРОГРАММЫ

### 6.1 Предмет проверки

Общее заключение о работоспособности инструментальных средств разработки геоинформационных Web-приложений GIS WebToolKit SE даётся после проверки результатов выполнения сценариев JavaScript с применением тестовых примеров и приложения GIS WebServer SE.

Перед проверкой необходимо установить на Web-сервере и настроить для использования картографический сервис GIS WebService SE и Web-приложение GIS WebServer SE.

Для подключения GIS WebToolKit SE необходимо, чтобы компоненты были загружены в браузер вместе с кодом страницы как обычные внешние JavaScript-файлы:

```
<script src="Scripts/wqueries.js" type="text/javascript"></script>
<script src="Scripts/wmtsqueries.js" type="text/javascript"></script>
<script src="Scripts/wfsqueries.js" type="text/javascript"></script>
<script src="Scripts/wmsqueries.js" type="text/javascript"></script>
```

### 6.2 Порядок проверки

#### 6.2.1 Порядок проверки выполнения операций протокола OGC WMTS

В процессе проверки выполняются запросы картографических данных и метаданных к сервису GIS WebService SE по протоколу OGC WMTS версии 1.0.0:

- GetCapabilities,
- GetTile,
- GetFeatureInfo.

Для выполнения запросов используются методы класса WmtsQueries.

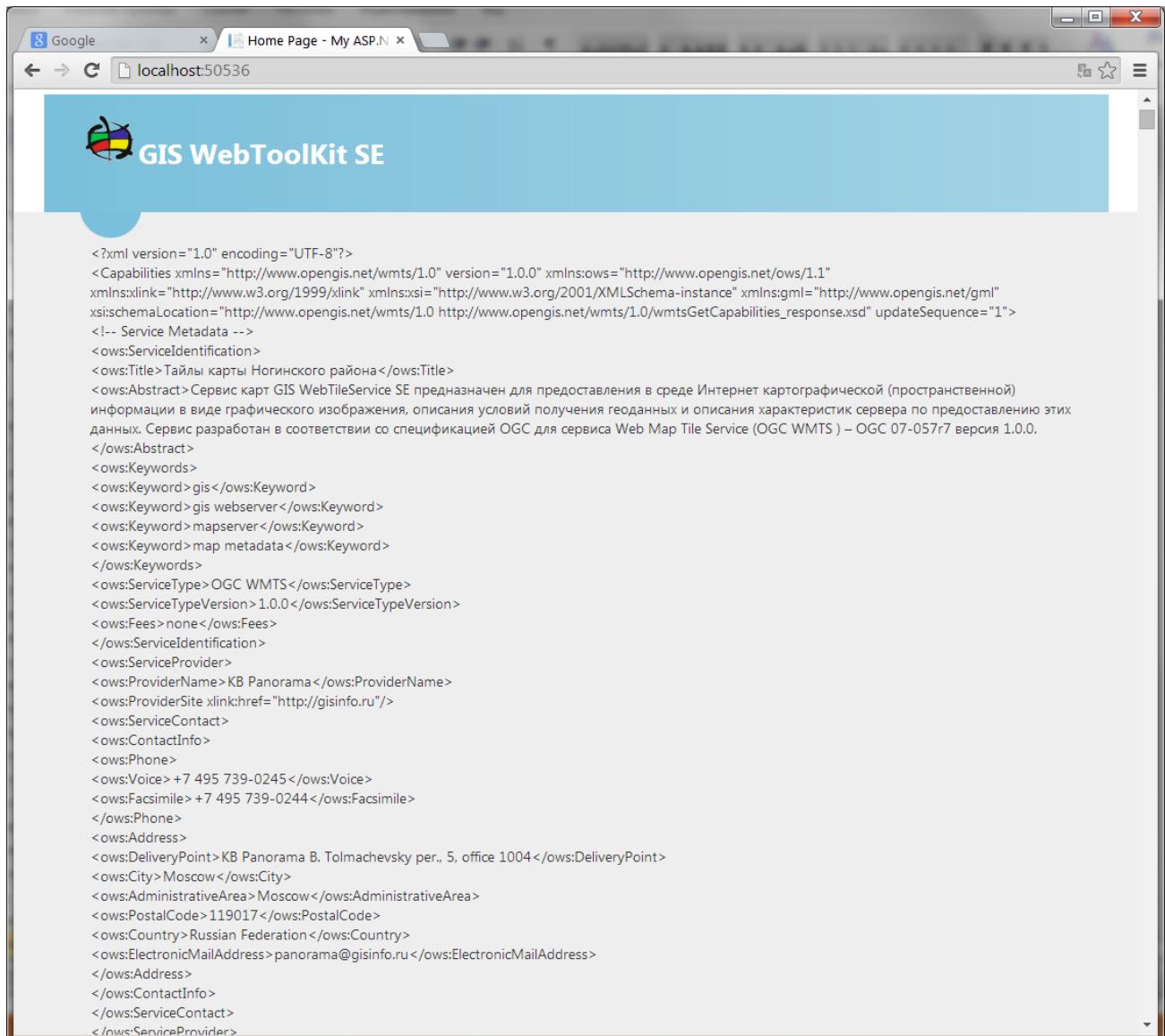
##### 6.2.1.1 Проверка получения метаданных об имеющихся на сервере слоях (картах) и возможных значениях параметров запросов – запрос GetCapabilities

Для получения метаданных необходимо выполнить метод **getcapabilities()** класса WmtsQueries. Тестовый скрипт testWmtsCapabilities имеет вид:

```
function onDataLoaded(response) {
    var elem = $get("dvMap");
    if (elem != null) elem.innerText = response;
}

function testWmtsCapabilities()
{
    var wmts = new WmtsQueries("http://gisserver.info/GISWebServiceSE/service.php");
    wmts.onDataLoad = onDataLoaded;
    wmts.getcapabilities();
}
```

**Результат:** в случае успешной проверки в браузер выводится XML-документ метаданных сервиса OGC WMTS:



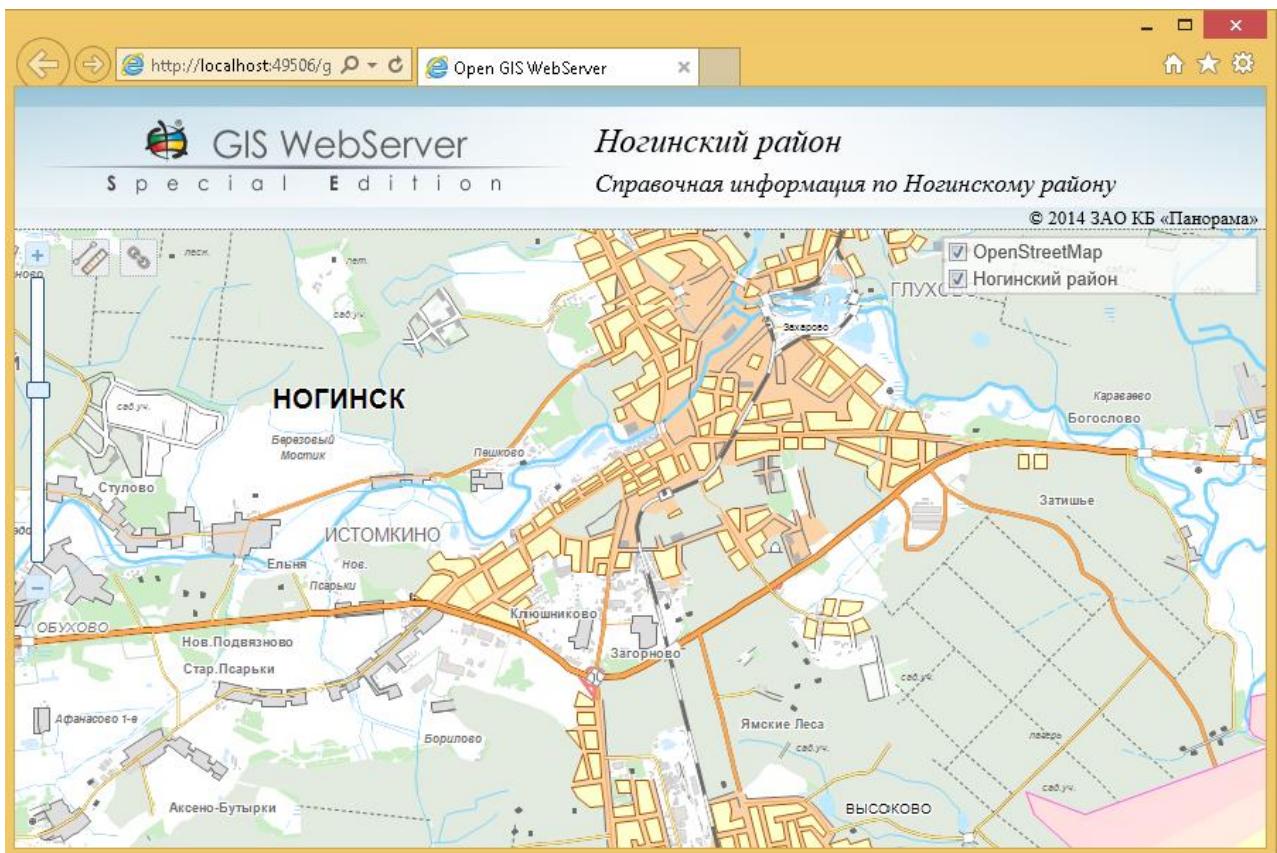
The screenshot shows a web browser window with the title "Home Page - My ASP.NET". The address bar displays "localhost:50536". The main content area features a blue header with the text "GIS WebToolKit SE" and a small globe icon. Below the header is a large block of XML code representing the service metadata.

```
<?xml version="1.0" encoding="UTF-8"?>
<Capabilities xmlns="http://www.opengis.net/wmts/1.0" version="1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gml="http://www.opengis.net/gml"
  xsi:schemaLocation="http://www.opengis.net/wmts/1.0 http://www.opengis.net/wmts/1.0/wmtsGetCapabilities_response.xsd" updateSequence="1">
  <!-- Service Metadata -->
  <ows:ServiceIdentification>
    <ows:Title>Тайлы карты Ногинского района</ows:Title>
    <ows:Abstract>Сервис карт GIS WebTileService SE предназначен для предоставления в среде Интернет картографической (пространственной) информации в виде графического изображения, описания условий получения геоданных и описания характеристик сервера по предоставлению этих данных. Сервис разработан в соответствии со спецификацией OGC для сервиса Web Map Tile Service (OGC WMTS) – OGC 07-057r7 версия 1.0.0.</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>gis</ows:Keyword>
      <ows:Keyword>gis webserver</ows:Keyword>
      <ows:Keyword>mapserver</ows:Keyword>
      <ows:Keyword>map metadata</ows:Keyword>
    </ows:Keywords>
    <ows:ServiceType>OGC WMTS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
    <ows:Fees>none</ows:Fees>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ProviderName>KB Panorama</ows:ProviderName>
    <ows:ProviderSite xlink:href="http://gisinfo.ru/"></ows:ProviderSite>
    <ows:ServiceContact>
      <ows>ContactInfo>
        <ows:Phone>
          <ows:Voice>+7 495 739-0245</ows:Voice>
          <ows:Facsimile>+7 495 739-0244</ows:Facsimile>
        </ows:Phone>
        <ows:Address>
          <ows:DeliveryPoint>KB Panorama B. Tolmachevsky per., 5, office 1004</ows:DeliveryPoint>
          <ows:City>Moscow</ows:City>
          <ows:AdministrativeArea>Moscow</ows:AdministrativeArea>
          <ows:PostalCode>119017</ows:PostalCode>
          <ows:Country>Russian Federation</ows:Country>
          <ows:ElectronicMailAddress>panorama@gisinfo.ru</ows:ElectronicMailAddress>
        </ows:Address>
        <ows:ContactInfo>
        </ows:ContactInfo>
      </ows>ContactInfo>
    </ows:ServiceContact>
  </ows:ServiceProvider>
```

6.2.1.2 Проверка получения изображений карты – запрос GetTile  
Для проверки выполнения запроса GetTile необходимо:  
– стартовать приложение GIS WebServer SE в браузере;  
– включить кнопку «Ногинский район» для отображения карты.

**Результат:**

В окне браузера отобразится карта «Ногинский район».



#### 6.2.1.3 Получение метаданных объектов карты в указанном местоположении - запрос GetFeatureInfo

Для получения метаданных объектов карты необходимо выполнить метод featureinfo класса WmtsQueries. Тестовый скрипт testGetFeatureInfo имеет вид:

```
var wmts = null;
function onDataLoaded(response) {
    var elem = $get("dvMap");
    if (elem != null) elem.innerHTML = response
}

function testGetFeatureInfo()
{
    wmts = new WmtsQueries("http://gisserver.info/GISWebServiceSE/service.php");
    wmts.onDataLoad = onDataLoaded;
    // параметры запроса
    var queryfeature = '{"layer": "0002", "tilematrixset": "GoogleMapsCompatible",' +
        '"style": "default", "format": "image/jpg", "tilematrix": "10", "tilerow":319, "tilecol": 621, ' +
        '"info_format": "text/html", "feature_count":5, "i":100,"j":50}';

    wmts.featureinfo(queryfeature);
}
```

#### Результат:

В окне браузера отобразятся метаданные объектов карты «Ногинский район».

The screenshot shows a Microsoft Internet Explorer window with the title "Home Page - My ASP.NET Application - Windows Internet Explorer". The address bar shows "http://localhost:50536/". The main content area is titled "GIS WebToolKit SE" and contains a table of object properties:

Объект	
Имя семантики	Значение семантики
НАЗВАНИЕ СОБСТВ.ТЕКСТ ПОДПИСИ	Иванисово
КОЛИЧЕСТВО ЖИТЕЛЕЙ ПО ШКАЛЕ	ОТ 100 ДО 500
Объект	
Имя семантики	Значение семантики
НАЗВАНИЕ СОБСТВ.ТЕКСТ ПОДПИСИ	Фрязево
КОЛИЧЕСТВО ЖИТЕЛЕЙ ПО ШКАЛЕ	ОТ 500 ДО 1 ТЫС.
Объект	
Имя семантики	Значение семантики
ТИП ВОДОТОКА ( ВОДОЕМА )	ПОСТОЯННЫЙ
АБСОЛЮТНАЯ ВЫСОТА	137.71
Объект	
Имя семантики	Значение семантики
ТИП ВОДОТОКА ( ВОДОЕМА )	ПОСТОЯННЫЙ
ШИРИНА ПО ШКАЛЕ	МЕНЕЕ 20 М
ПРИЗНАК СУДОХОДСТВА	НЕСУДОХОДНЫЙ
НАЗВАНИЕ СОБСТВ.ТЕКСТ ПОДПИСИ	Вохонка
Объект	
Имя семантики	Значение семантики
СОСТОЯНИЕ	ДЕЙСТВУЮЩИЙ
МАТЕРИАЛ ПОКРЫТИЯ	АСФАЛЬТ(АСФАЛЬТОБЕТОН)

### 6.2.2 Порядок проверки выполнения операций протокола OGC WFS

В процессе проверки выполняются запросы картографических данных и метаданных к сервису GIS WebService SE по протоколу OGC WFS версии 2.0 для получения векторной и атрибутивной информации об объектах карты. Векторные данные объектов представляются в формате GML 3.2 (ISO 19136:2007). Проверяется выполнение запросов OGC WFS:

- GetCapabilities,
- DescribeFeatureType,
- GetFeature,
- ListStoredQueries,
- Transaction.

Для выполнения запросов используются методы класса WfsQueries.

Входные данные для методов изменения объектов (транзакций) имеют кодировку XML/GML в соответствии с прикладной XSSD-схемой топомар.xsd.

Выходные данные методов транзакций содержат результат выполнения в виде XML.

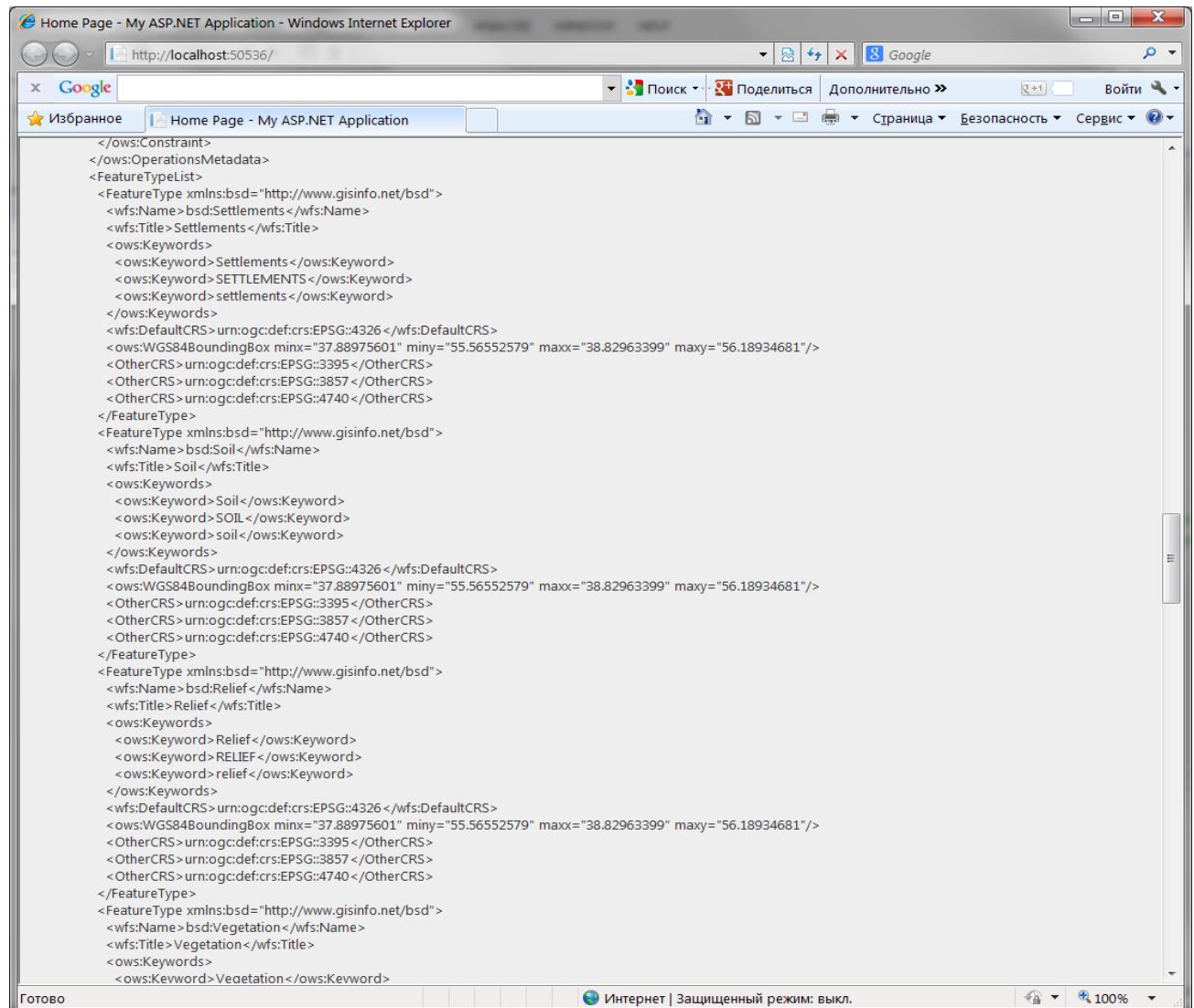
#### 6.2.2.1 Проверка получения метаданных об имеющихся типах объектов в базе пространственных данных и доступных операциях сервиса – запрос GetCapabilities

Для получения метаданных необходимо выполнить метод `getfeaturetypenamelist()` класса WfsQueries. Тестовый скрипт testWfsCapabilities имеет вид:

```
function onDataLoaded(response) {
    // ...
}

function testWfsCapabilities()
{
    var wfs = new WfsQueries("http://gisserver.info/GISWebServiceSE/service.php");
    // назначить обработчик для получения данных от сервера
    wfs.onDataLoad = onDataLoaded;
    // выполнить запрос
    wfs.getfeaturetypenamelist ();
}
```

**Результат:** в случае успешной проверки в браузер выводится XML-документ метаданных сервиса OGC WFS:



The screenshot shows a Microsoft Internet Explorer window with the title "Home Page - My ASP.NET Application - Windows Internet Explorer". The address bar contains "http://localhost:50536/". The main content area displays an XML document representing metadata for a WFS service. The XML includes various feature types such as Settlements, Soil, Relief, and Vegetation, each with their respective names, titles, keywords, and coordinate reference systems (CRS). The XML also includes sections for operations metadata and constraints.

```
</ows:Constraint>
</ows:OperationsMetadata>
<FeatureTypeList>
<FeatureType xmlns:bsd="http://www.gisinfo.net/bsd">
<wfs:Name>bsd:Setlements</wfs:Name>
<wfs:Title>Settlements</wfs:Title>
<ows:Keywords>
<ows:Keyword>Settlements</ows:Keyword>
<ows:Keyword>SETLEMENTS</ows:Keyword>
<ows:Keyword>settlements</ows:Keyword>
</ows:Keywords>
<wfs:DefaultCRS>urn:ogc:def:crs:EPSG::4326</wfs:DefaultCRS>
<ows:WGS84BoundingBox minx="37.88975601" miny="55.56552579" maxx="38.82963399" maxy="56.18934681"/>
<OtherCRS>urn:ogc:def:crs:EPSG::3395</OtherCRS>
<OtherCRS>urn:ogc:def:crs:EPSG::3857</OtherCRS>
<OtherCRS>urn:ogc:def:crs:EPSG::4740</OtherCRS>
</FeatureType>
<FeatureType xmlns:bsd="http://www.gisinfo.net/bsd">
<wfs:Name>bsd:Soil</wfs:Name>
<wfs:Title>Soil</wfs:Title>
<ows:Keywords>
<ows:Keyword>Soil</ows:Keyword>
<ows:Keyword>SOIL</ows:Keyword>
<ows:Keyword>soil</ows:Keyword>
</ows:Keywords>
<wfs:DefaultCRS>urn:ogc:def:crs:EPSG::4326</wfs:DefaultCRS>
<ows:WGS84BoundingBox minx="37.88975601" miny="55.56552579" maxx="38.82963399" maxy="56.18934681"/>
<OtherCRS>urn:ogc:def:crs:EPSG::3395</OtherCRS>
<OtherCRS>urn:ogc:def:crs:EPSG::3857</OtherCRS>
<OtherCRS>urn:ogc:def:crs:EPSG::4740</OtherCRS>
</FeatureType>
<FeatureType xmlns:bsd="http://www.gisinfo.net/bsd">
<wfs:Name>bsd:Relief</wfs:Name>
<wfs:Title>Relief</wfs:Title>
<ows:Keywords>
<ows:Keyword>Relief</ows:Keyword>
<ows:Keyword>RELIEF</ows:Keyword>
<ows:Keyword>relief</ows:Keyword>
</ows:Keywords>
<wfs:DefaultCRS>urn:ogc:def:crs:EPSG::4326</wfs:DefaultCRS>
<ows:WGS84BoundingBox minx="37.88975601" miny="55.56552579" maxx="38.82963399" maxy="56.18934681"/>
<OtherCRS>urn:ogc:def:crs:EPSG::3395</OtherCRS>
<OtherCRS>urn:ogc:def:crs:EPSG::3857</OtherCRS>
<OtherCRS>urn:ogc:def:crs:EPSG::4740</OtherCRS>
</FeatureType>
<FeatureType xmlns:bsd="http://www.gisinfo.net/bsd">
<wfs:Name>bsd:Vegetation</wfs:Name>
<wfs:Title>Vegetation</wfs:Title>
<ows:Keywords>
<ows:Keyword>Veaeatation</ows:Keyword>
```

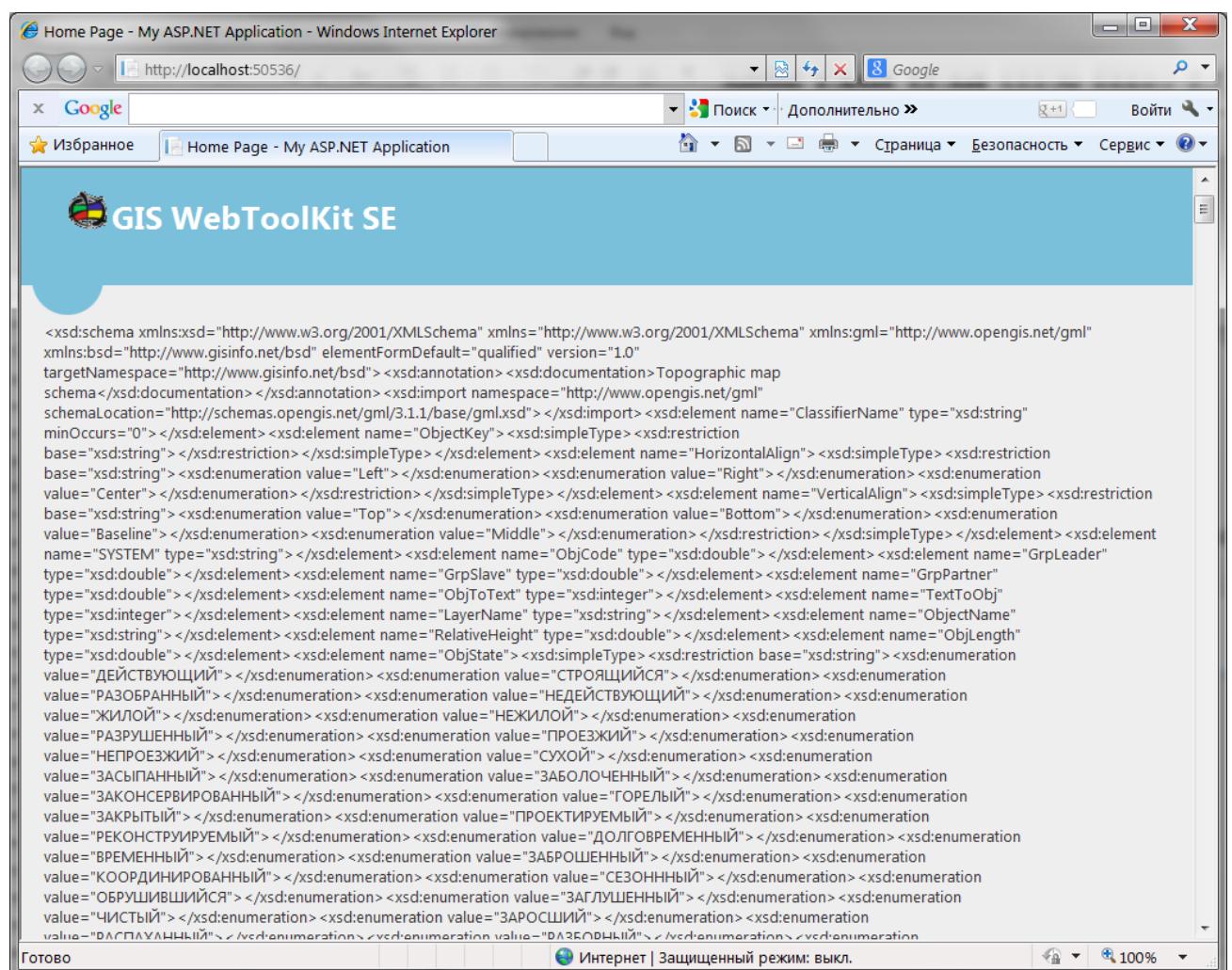
#### 6.2.2.2 Проверка получения XSD-схемы описания типов объектов – запрос DescribeFeatureType

Для получения XSD-схемы необходимо выполнить метод **featuretype()** класса Wfs-Queries.

Тестовый скрипт testWfsDescribeFeatureType имеет вид:

```
function onDataLoaded(response) {  
    // ...  
}  
  
function testWfsDescribeFeatureType ()  
{  
    var wfs = new WfsQueries("http://gisserver.info/GISWebServiceSE/service.php");  
    // назначить обработчик для получения данных от сервера  
    wfs.onDataLoad = onDataLoaded;  
  
    // выполнить запрос  
    wfs.featuretype ();  
}
```

**Результат:** в случае успешной проверки в браузер выводится XML-документ XSD-схемы набора данных:



The screenshot shows a Microsoft Internet Explorer window displaying an XML schema document. The title bar reads "Home Page - My ASP.NET Application - Windows Internet Explorer". The address bar shows "http://localhost:50536/". The main content area displays the XML code for the "Topographic map" schema, which includes definitions for elements like ClassifierName, ObjCode, GrpLeader, and various enumeration values such as Left, Right, Center, Top, Bottom, Middle, etc. The XML is well-formatted with line breaks and indentations.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"  
    xmlns:bsd="http://www.gisinfo.net/bsd" elementFormDefault="qualified" version="1.0"  
    targetNamespace="http://www.gisinfo.net/bsd" ><xsd:annotation><xsd:documentation>Topographic map  
    schema</xsd:documentation></xsd:annotation><xsd:import namespace="http://www.opengis.net/gml"  
    schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd" /></xsd:import><xsd:element name="ClassifierName" type="xsd:string"  
    minOccurs="0" /></xsd:element><xsd:element name="ObjectKey"><xsd:simpleType><xsd:restriction  
    base="xsd:string"></xsd:restriction></xsd:simpleType></xsd:element><xsd:element name="HorizontalAlign"><xsd:simpleType><xsd:restriction  
    base="xsd:string"></xsd:restriction></xsd:simpleType></xsd:element><xsd:element name="VerticalAlign"><xsd:simpleType><xsd:restriction  
    base="xsd:string"></xsd:restriction></xsd:simpleType></xsd:element><xsd:element name="Baseline"><xsd:enumeration value="Top" /></xsd:enumeration><xsd:enumeration value="Bottom" /></xsd:enumeration><xsd:enumeration  
    value="Middle" /></xsd:enumeration><xsd:enumeration value="Center" /></xsd:enumeration></xsd:restriction></xsd:simpleType></xsd:element><xsd:element  
    name="SYSTEM" type="xsd:string" /></xsd:element><xsd:element name="ObjCode" type="xsd:double" /></xsd:element><xsd:element name="GrpLeader"  
    type="xsd:double" /></xsd:element><xsd:element name="GrpSlave" type="xsd:double" /></xsd:element><xsd:element name="GrpPartner"  
    type="xsd:double" /></xsd:element><xsd:element name="ObjToText" type="xsd:integer" /></xsd:element><xsd:element name="TextToObj"  
    type="xsd:integer" /></xsd:element><xsd:element name="LayerName" type="xsd:string" /></xsd:element><xsd:element name="ObjectName"  
    type="xsd:string" /></xsd:element><xsd:element name="RelativeHeight" type="xsd:double" /></xsd:element><xsd:element name="ObjLength"  
    type="xsd:double" /></xsd:element><xsd:element name="ObjState" type="xsd:simpleType" /><xsd:restriction base="xsd:string" /></xsd:enumeration  
    value="ДЕЙСТВУЮЩИЙ" /></xsd:enumeration><xsd:enumeration value="СТРОЯЩИЙСЯ" /></xsd:enumeration><xsd:enumeration  
    value="РАЗОБРАННЫЙ" /></xsd:enumeration><xsd:enumeration value="НЕДЕЙСТВУЮЩИЙ" /></xsd:enumeration><xsd:enumeration  
    value="ЖИЛОЙ" /></xsd:enumeration><xsd:enumeration value="НЕЖИЛОЙ" /></xsd:enumeration><xsd:enumeration  
    value="РАЗРУШЕННЫЙ" /></xsd:enumeration><xsd:enumeration value="ПРОЕЗЖИЙ" /></xsd:enumeration><xsd:enumeration  
    value="НЕПРОЕЗЖИЙ" /></xsd:enumeration><xsd:enumeration value="СУХОЙ" /></xsd:enumeration><xsd:enumeration  
    value="ЗАСЫПАННЫЙ" /></xsd:enumeration><xsd:enumeration value="ЗАБОЛОЧЕННЫЙ" /></xsd:enumeration><xsd:enumeration  
    value="ЗАКОНСЕРИВОВАННЫЙ" /></xsd:enumeration><xsd:enumeration value="ГОРЕЛЫЙ" /></xsd:enumeration><xsd:enumeration  
    value="ЗАКРЫТЫЙ" /></xsd:enumeration><xsd:enumeration value="ПРОЕКТИРУЕМЫЙ" /></xsd:enumeration><xsd:enumeration  
    value="РЕКОНСТРУИРУЕМЫЙ" /></xsd:enumeration><xsd:enumeration value="ДОЛГОВРЕМЕННЫЙ" /></xsd:enumeration><xsd:enumeration  
    value="ВРЕМЕННЫЙ" /></xsd:enumeration><xsd:enumeration value="ЗАБРОШЕННЫЙ" /></xsd:enumeration><xsd:enumeration  
    value="КООРДИНИРОВАННЫЙ" /></xsd:enumeration><xsd:enumeration value="СЕЗОННЫЙ" /></xsd:enumeration><xsd:enumeration  
    value="ОБРУШИВШИЙСЯ" /></xsd:enumeration><xsd:enumeration value="ЗАГЛУШЕННЫЙ" /></xsd:enumeration><xsd:enumeration  
    value="ЧИСТЫЙ" /></xsd:enumeration><xsd:enumeration value="ЗАРОСШИЙ" /></xsd:enumeration><xsd:enumeration  
    value="ДАГЛАУЛЧИЙ" /></xsd:enumeration><xsd:enumeration value="ДАСПОНЧИЙ" /></xsd:enumeration><xsd:enumeration  
    value="ДАСПОНЧИЙ" /></xsd:enumeration>
```

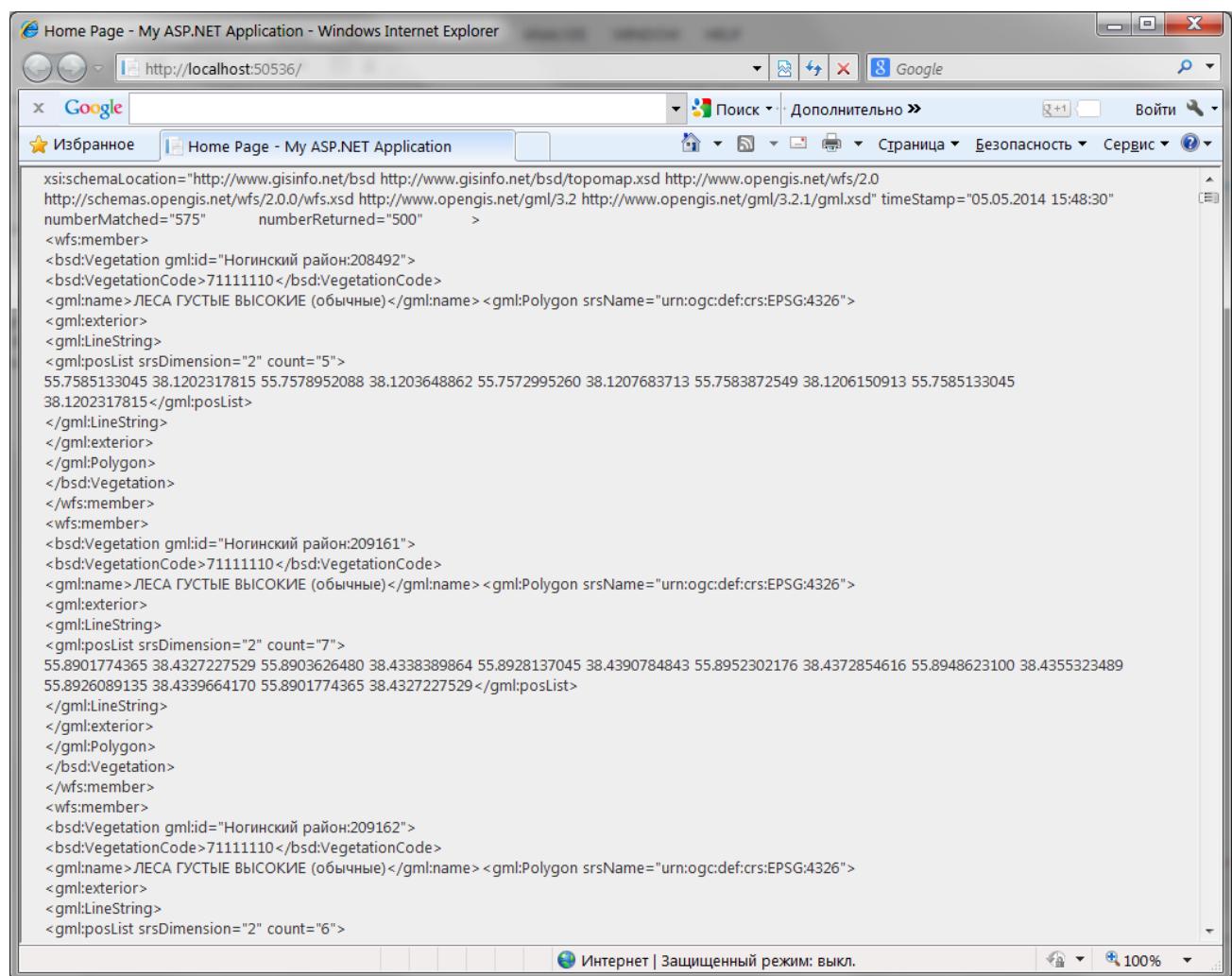
#### 6.2.2.3 Проверка получения векторных данных объектов – запрос GetFeature

Для проверки получения векторных данных объектов карты необходимо выполнить метод **feature()** класса WfsQueries.

В тестовом примере запрашиваются объекты типа bsd:Vegetation не более 500. Тестовый скрипт имеет вид:

```
var wfs = new WfsQueries("http://gisserver.info/GISWebServiceSE/service.php ");
wfs.onDataLoad = onDataLoaded;
var queryfeature = '{ "TypeNames": "bsd:Vegetation", "count": 500, "startIndex": "0" }';
wfs.feature(queryfeature);
```

**Результат:** в случае успешной проверки в браузер выводится XML-документ, содержащий векторные данные объектов типа bsd:Vegetation набора данных в формате GML:



The screenshot shows a Microsoft Internet Explorer window with the title 'Home Page - My ASP.NET Application - Windows Internet Explorer'. The address bar displays 'http://localhost:50536/'. The main content area of the browser shows a large block of XML code representing GML (Geographic Markup Language) data. The XML describes several polygons representing vegetation areas, specifically 'ЛЕСА ГУСТЫЕ ВЫСОКИЕ (обычные)' (Thick High Common Forests). The XML includes schemaLocation declarations, member elements, and detailed geometry definitions like exterior and interior boundaries and coordinate lists. At the bottom of the browser window, there is a status bar with the text 'Интернет | Защищенный режим: выкл.' (Internet | Protected mode: off) and a zoom level indicator of '100%'. The browser interface includes standard toolbar icons and a menu bar with Russian labels.

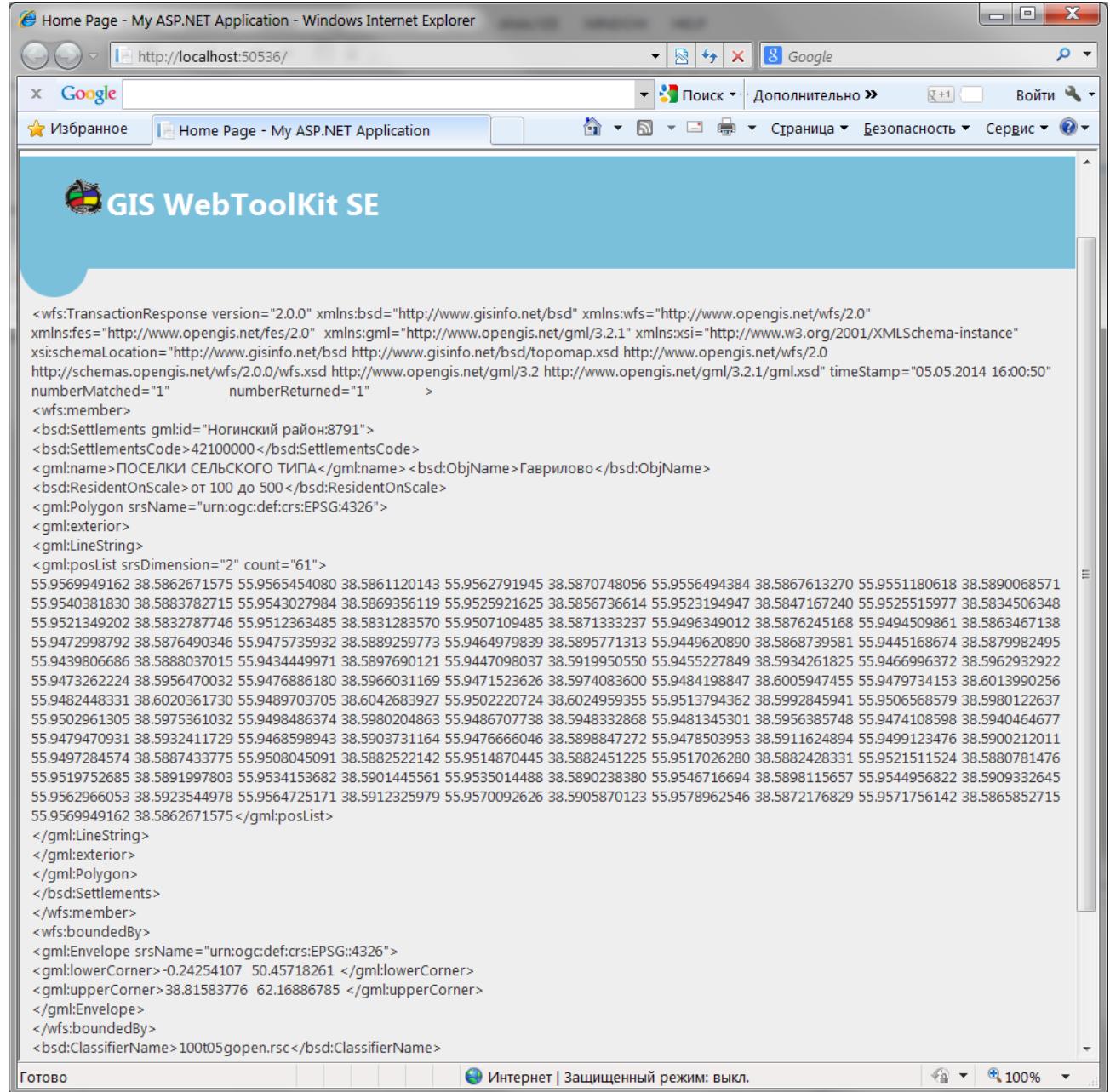
#### 6.2.2.4 Проверка получения векторных и атрибутивных данных одного объекта набора данных по идентификатору – запрос GetFeatureById

Для проверки получения векторных данных объекта карты по идентификатору объекта необходимо выполнить метод **featurebyid()** класса WfsQueries.

В тестовом примере запрашивается объект набора данных с идентификатором "Ногинский район:8791". Тестовый скрипт имеет вид:

```
var wfs = new WfsQueries("http://gisserver.info/GISWebServiceSE/service.php ");
wfs.onDataLoad = onDataLoaded;
wfs.featurebyid("Ногинский район:8791");
```

**Результат:** в случае успешной проверки в браузер выводится XML-документ, содержащий векторные данные одного объекта в формате GML, чей идентификатор совпал со значением *id* "Ногинский район:8791":



The screenshot shows a Microsoft Internet Explorer window with the title "Home Page - My ASP.NET Application - Windows Internet Explorer". The address bar displays "http://localhost:50536/". The main content area shows the "GIS WebToolKit SE" logo and an XML document. The XML content is as follows:

```
<wfs:TransactionResponse version="2.0.0" xmlns:bsd="http://www.gisinfo.net/bsd" xmlns:wfs="http://www.opengis.net/wfs/2.0"
xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:gml="http://www.opengis.net/gml/3.2.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.gisinfo.net/bsd http://www.gisinfo.net/bsd/topomap.xsd http://www.opengis.net/wfs/2.0
http://schemas.opengis.net/wfs/2.0/wfs.xsd http://www.opengis.net/gml/3.2 http://www.opengis.net/gml/3.2.1/gml.xsd" timeStamp="05.05.2014 16:00:50"
numberMatched="1" numberReturned="1" >
<wfs:member>
<bsd:Settlements gml:id="Ногинский район:8791">
<bsd:SettlementsCode>42100000</bsd:SettlementsCode>
<gml:name>ПОСЕЛКИ СЕЛЬСКОГО ТИПА</gml:name><bsd:ObjName>Гаврилово</bsd:ObjName>
<bsd:ResidentOnScale>от 100 до 500</bsd:ResidentOnScale>
<gml:Polygon srsName="urn:ogc:def:crs:EPSG:4326">
<gml:exterior>
<gml:LineString>
<gml:posList srsDimension="2" count="61">
55.9569949162 38.5862671575 55.9565454080 38.5861120143 55.9562791945 38.5870748056 55.9556494384 38.5867613270 55.9551180618 38.5890068571
55.9540381830 38.5883782715 55.9543027984 38.5869356119 55.9525921625 38.5856736614 55.9523194947 38.5847167240 55.9525515977 38.5834506348
55.9521349202 38.5832787746 55.9512363485 38.5831283570 55.9507109485 38.5871333237 55.9496349012 38.5876245168 55.9494509861 38.5863467138
55.9472998792 38.5876490346 55.9475735932 38.5889259773 55.9464979839 38.5895771313 55.944620890 38.5868739581 55.9445168674 38.5879982495
55.9439806686 38.5888037015 55.9434449971 38.5897690121 55.9447098037 38.5919950550 55.9455227849 38.5934261825 55.9466996372 38.5962932922
55.9473262224 38.5956470032 55.9476886180 38.5966031169 55.9471523626 38.5974083600 55.9484198847 38.6005947455 55.9479734153 38.6013990256
55.9482448331 38.6020361730 55.9489703705 38.6042683927 55.9502220724 38.6024959355 55.9513794362 38.5992845941 55.9506568579 38.5980122637
55.9502961305 38.5975361032 55.9498486374 38.5980204863 55.9486707738 38.5948332868 55.9481345301 38.5956385748 55.9474108598 38.5940464677
55.9479470931 38.5932411729 55.9468598943 38.5903731164 55.9476666046 38.5898847272 55.9478503953 38.5911624894 55.9499123476 38.5900212011
55.9497284574 38.5887433775 55.9508045091 38.5882522142 55.9514870445 38.5882451225 55.9517026280 38.5882428331 55.9521511524 38.5880781476
55.9519752685 38.5891997803 55.9534153682 38.5901445561 55.9535014488 38.5890238380 55.9546716694 38.5898115657 55.9544956822 38.5909332645
55.9562966053 38.5923544978 55.9564725171 38.5912325979 55.9570092626 38.5905870123 55.9578962546 38.5872176829 55.9571756142 38.5865852715
55.9569949162 38.5862671575</gml:posList>
</gml:LineString>
</gml:exterior>
</gml:Polygon>
</bsd:Settlements>
</wfs:member>
<wfs:boundedBy>
<gml:Envelope srsName="urn:ogc:def:crs:EPSG::4326">
<gml:lowerCorner>-0.24254107 50.45718261 </gml:lowerCorner>
<gml:upperCorner>38.81583776 62.16886785 </gml:upperCorner>
</gml:Envelope>
</wfs:boundedBy>
<bsd:ClassifierName>100t05gopen.rsc</bsd:ClassifierName>
```

#### 6.2.2.5 Проверка выполнения транзакций – запрос Transaction

Для проверки создания, изменения или удаления объектов набора данных необходимо выполнить метод **transaction(xmlaction)** класса **WfsQueries**.

Входные данные метода содержат XML-запрос операции транзакции в соответствии со стандартом OGC WFS 2.0. Операция транзакции содержит команду для создания, замены или удаления объектов набора данных.

Для проверки создания объекта при помощи запроса транзакции **wfs:Transaction/wfs:Insert** необходимо выполнить тестовый скрипт вида:

```
var wfs = new WfsQueries("http://gisserver.info/GISWebServiceSE/service.php ");
wfs.onDataLoad = onDataLoaded;
wfs.transaction(xmlInsert);
```

Где *xmlInsert* - тестовый XML-запрос для создания объекта имеет вид:

```
<?xml version="1.0" encoding="utf-8" ?
<wfs:Transaction version="2.0.0" service="WFS" handle="Transaction 1"
xmlns:bsd="http://www.gisinfo.net/bsd" xmlns:wfs="http://www.opengis.net/wfs/2.0"
xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.gisinfo.net/bsd http://www.gisinfo.net/bsd/topomap.xsd
http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
http://www.opengis.net/gml/3.2 http://www.opengis.net/gml/3.2.1/gml.xsd" >
<wfs:Insert handle="Ins1">
    <bsd:Vegetation gml:id="Ногинский район">
        <bsd:VegetationCode>71132100</bsd:VegetationCode>
        <gml:name>ДЕРЕВЬЯ (не имеющие значения ориентиров)</gml:name>
        <gml:Point srsName="urn:ogc:def:crs:EPSG:4326">
            <gml:pos srsDimension="3">55.7531215306 38.3349540667 -111111.0</gml:pos>
        </gml:Point>
    </bsd:Vegetation>
</wfs:Insert>
</wfs:Transaction>
```

**Результат:** в случае успешной проверки в браузер выводится XML-документ, содержащий ответ выполненной транзакции **wfs:Transaction/wfs:Insert**:

```
<?xml version="1.0" encoding="utf-8"?>
<wfs:TransactionResponse version="2.0.0" xmlns:bsd="http://www.gisinfo.net/bsd" xmlns:wfs="http://www.opengis.net/wfs/2.0"
xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:gml="http://www.opengis.net/gml/3.2.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.gisinfo.net/bsd http://www.gisinfo.net/bsd/topomap.xsd http://www.opengis.net/wfs/2.0
http://schemas.opengis.net/wfs/2.0.0/wfs.xsd http://www.opengis.net/gml/3.2 http://www.opengis.net/gml/3.2.1/gml.xsd" >
<wfs:TransactionSummary>
<wfs:totalInserted>1</wfs:totalInserted> <wfs:totalReplaced>0</wfs:totalReplaced> <wfs:totalUpdated>0</wfs:totalUpdated> </wfs:TransactionSummary>
<wfs:InsertResults>
<wfs:Feature handle="Insert 0">
<wfs:ResourceId rid="Ногинский район:4861"/>
</wfs:Feature>
</wfs:InsertResults>
</wfs:TransactionResponse>
```

Для проверки замены существующего объекта набора данных при помощи запроса транзакции необходимо выполнить команду транзакции **wfs:Transaction/wfs:Replace**.

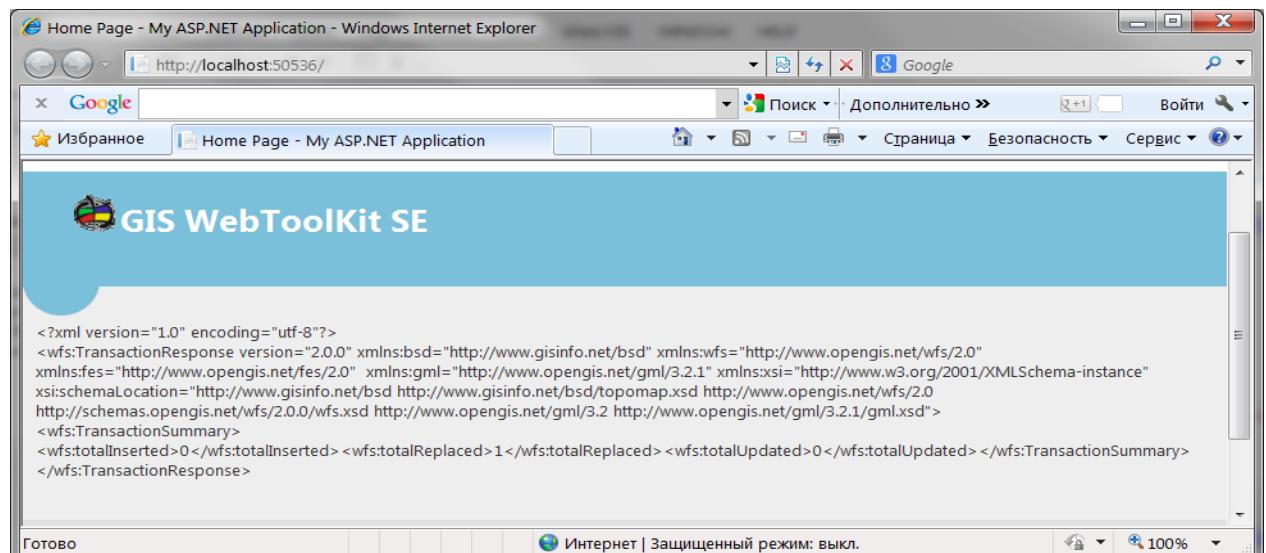
Тестовый пример выполняет замену объекта набора данных с идентификатором "Ногинский район:209263", XML-запрос для замещения объекта имеет вид:

```
<?xml version="1.0" encoding="utf-8" ?>
<wfs:Transaction version="2.0.0" service="WFS" handle="Transaction 2"
  xmlns:bsd="http://www.gisinfo.net/bsd" xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.gisinfo.net/bsd http://www.gisinfo.net/bsd/topomap.xsd
  http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
  http://www.opengis.net/gml/3.2 http://www.opengis.net/gml/3.2.1/gml.xsd" >
<wfs:Replace handle="replace 01">
  <fes:Filter><fes:ResourceId rid="Ногинский район:209263" /></fes:Filter>
  <bsd:Vegetation gml:id="id209263">
    <bsd:VegetationCode>71100000</bsd:VegetationCode>
    <gml:name>ДРЕВЕЧНАЯ РАСТИТЕЛЬНОСТЬ</gml:name>
    <gml:Polygon srsName="urn:ogc:def:crs:EPSG:4326">
      <gml:exterior><gml:LineString>
        <gml:posList srsDimension="2" count="9">55.7509419844 38.1351148613
55.7494491867 38.1402435206 55.7491871219 38.1413639697 55.7509036941 38.1429183731
55.7512649677 38.1432288919 55.7522287416 38.1395449989 55.7524041563 38.1389041549
55.7524728304 38.1357177285 55.7509419844 38.1351148613
        </gml:posList></gml:LineString>
      </gml:exterior>
    </gml:Polygon> </bsd:Vegetation>
  </wfs:Replace>
</wfs:Transaction>
```

Тестовый скрипт имеет вид:

```
var wfs = new WfsQueries("http://192.168.0.22/giswebse/wms.php");
wfs.onDataLoad = onDataLoaded;
wfs.transaction(xmlReplace);
```

**Результат:** в случае успешной проверки в браузер выводится XML-документ, содержащий ответ выполненной транзакции **wfs:Transaction/wfs:Replace**:



The screenshot shows a Microsoft Internet Explorer window with the title bar "Home Page - My ASP.NET Application - Windows Internet Explorer". The address bar shows "http://localhost:50536/". The main content area displays the XML response from the transaction. At the top of the XML is the header:

```
<?xml version="1.0" encoding="utf-8"?>
<wfs:TransactionResponse version="2.0.0" xmlns:bsd="http://www.gisinfo.net/bsd" xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:gml="http://www.opengis.net/gml/3.2.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.gisinfo.net/bsd http://www.gisinfo.net/bsd/topomap.xsd http://www.opengis.net/wfs/2.0
  http://schemas.opengis.net/wfs/2.0.0/wfs.xsd http://www.opengis.net/gml/3.2 http://www.opengis.net/gml/3.2.1/gml.xsd" >
```

Below the header, the XML continues with the transaction summary:

```
<wfs:TransactionSummary>
  <wfs:totalInserted>0</wfs:totalInserted> <wfs:totalReplaced>1</wfs:totalReplaced> <wfs:totalUpdated>0</wfs:totalUpdated> </wfs:TransactionSummary>
</wfs:TransactionResponse>
```

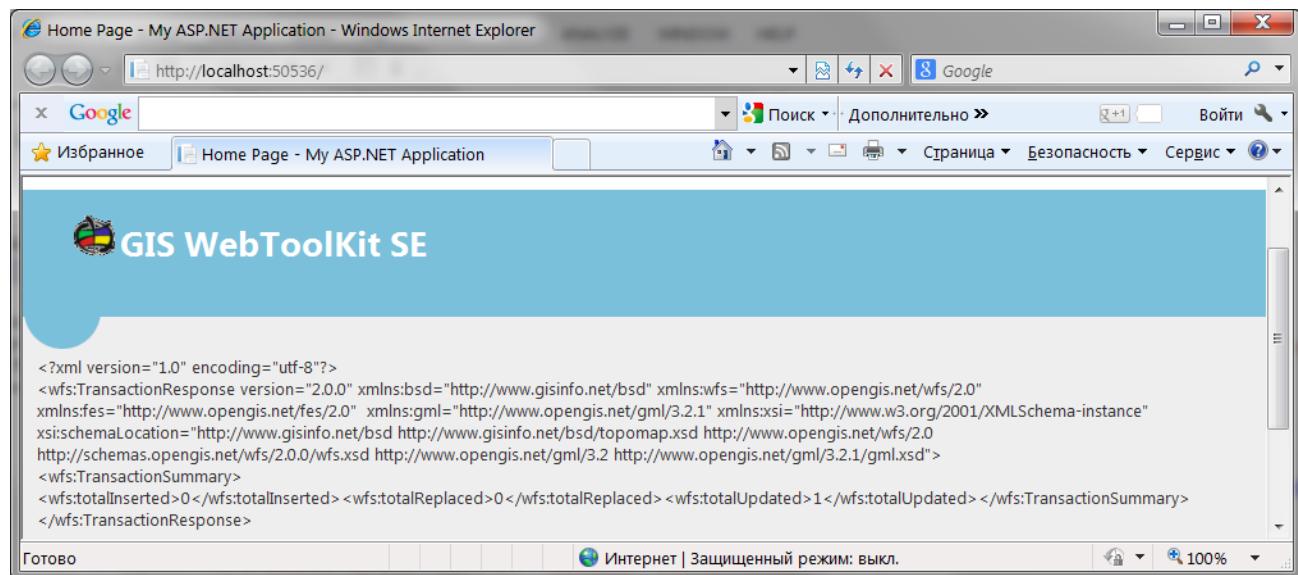
At the bottom of the browser window, there are status bars: "Готово" (Ready), "Интернет | Защищенный режим: выкл." (Internet | Protected mode: off), and a zoom level of "100%".

Для проверки удаления объекта из набора данных при помощи запроса транзакции необходимо выполнить команду транзакции **wfs:Transaction/wfs:Delete**.

Тестовый пример выполняет удаление объекта набора данных с идентификатором "Ногинский район:8791", XML-запрос транзакции для удаления объекта имеет вид:

```
<?xml version="1.0" encoding="utf-8" ?>
<wfs:Transaction version="2.0.0" service="WFS" handle="Transaction 3"
  xmlns:bsd="http://www.gisinfo.net/bsd" xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.gisinfo.net/bsd http://www.gisinfo.net/bsd/topomap.xsd
  http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
  http://www.opengis.net/gml/3.2 http://www.opengis.net/gml/3.2.1/gml.xsd" >
  <wfs:Delete handle="delete 01">
    <fes:Filter><fes:ResourceId rid="Ногинский район:8791" /></fes:Filter>
  </wfs:Delete>
</wfs:Transaction>
```

**Результат:** в случае успешной проверки в браузер выводится XML-документ, содержащий ответ выполненной транзакции **wfs:Transaction/wfs:Delete**:



### 6.2.3 Порядок проверки выполнения операций протокола OGC WMS

В процессе проверки выполняются запросы к сервису GIS WebService SE для получения изображения карты, для получения метаданных объектов карты и метаданных о доступных картах и возможностях сервера. Запросы выполняются по протоколу OGC WMS версии 1.3.0. Проверяются операции:

- GetCapabilities,
- GetMap,
- GetFeatureInfo.

Для выполнения запросов используются методы класса WmsQueries.

#### 6.2.3.1 Проверка получения метаданных об имеющихся на сервере картах и возможных значениях параметров запросов – запрос GetCapabilities

Для получения метаданных необходимо выполнить метод **getcapabilities()** класса WmsQueries. Тестовый скрипт имеет вид:

```
function onDataLoaded(response) {
    var elem = $get("dvMap");
    if (elem != null) elem.innerText = response;
}
wms = new WmsQueries("http://gisserver.info/GISWebService/service.php");
wms.onDataLoad = onDataLoaded;

wms.getcapabilities();
```

**Результат:** в случае успешной проверки в браузер выводится XML-документ метаданных сервиса OGC WMS:



The screenshot shows a Microsoft Internet Explorer window with the title bar "ГИС ПАНОРАМА :: ГИС" and the address bar "gisserver.info/GISWebS". The main content area displays an XML document representing the WMS capabilities. The XML includes various layers, coordinate reference systems (CRS), geographic bounding boxes, and metadata URLs. A scroll bar is visible on the right side of the browser window.

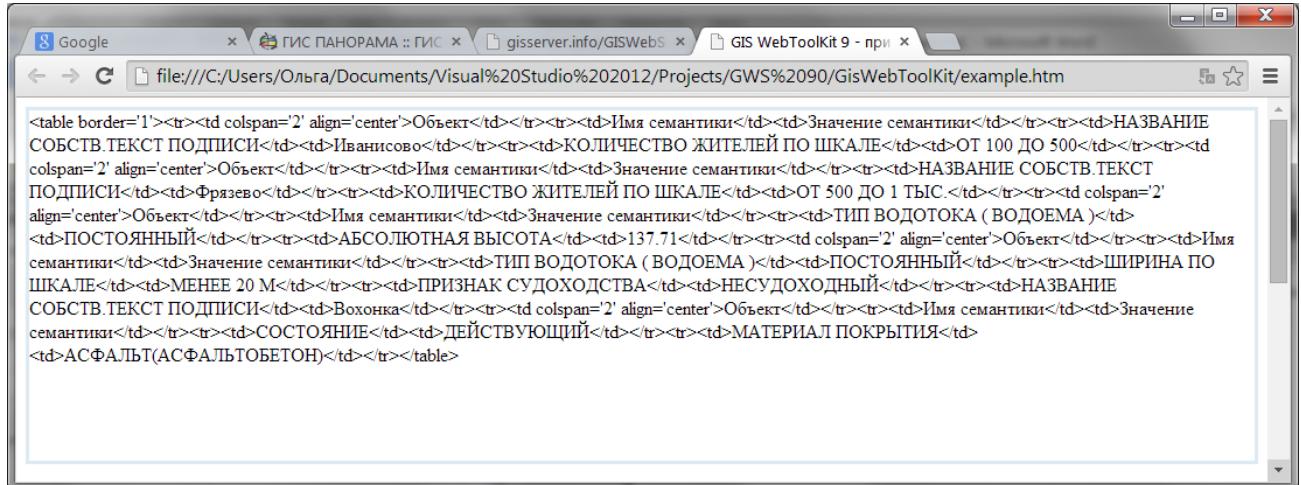
```
<CRS>EPSG:3857</CRS>
<CRS>EPSG:4740</CRS>
<CRS>EPSG:4326</CRS>
<Layer>
<Title>Ногинск</Title>
<Name>Noginsk</Name>
<CRS>EPSG:3395</CRS>
<CRS>EPSG:3857</CRS>
<CRS>EPSG:4740</CRS>
<CRS>EPSG:4326</CRS>
<EX_GeographicBoundingBox>
<westBoundLongitude>37.81636565</westBoundLongitude>
<eastBoundLongitude>38.89944438</eastBoundLongitude>
<southBoundLatitude>55.56608978</southBoundLatitude>
<northBoundLatitude>56.20480712</northBoundLatitude>
</EX_GeographicBoundingBox>
<MetadataURL type="ISO:19115:2003">
<BoundingBox CRS="EPSG:3395" minx="4209698.567" miny="7437247.162" maxx="4330266.341" maxy="7563757.536"/>
<BoundingBox CRS="EPSG:3857" minx="4209698.567" miny="7472516.909" maxx="4330266.341" maxy="7599295.466"/>
<BoundingBox CRS="EPSG:4740" minx="37.81636115" miny="55.56608672" maxx="38.89943974" maxy="56.20480411"/>
<BoundingBox CRS="EPSG:4326" minx="37.81636565" miny="55.56608978" maxx="38.89944438" maxy="56.20480712"/>
<Layer>
<Layer>
<Capability>
</WMS_Capabilities>
```

#### 6.2.3.2 Проверка получения метаданных объектов карты в указанном местоположении - запрос GetFeatureInfo

Для получения метаданных объектов карты необходимо выполнить метод **featureinfo()** класса WmsQueries. Тестовый скрипт wmsfeatureinfo() имеет вид:

```
function wmsfeatureinfo() {
    var queryfeature =
    {"layers":"Noginsk","crs":"EPSG:3395","style":"default","format":"image/jpg","bbox":4209698.567,7437247.162,4330266.341,7563757.536,"width":"256","height":"256","query_layers":"0002","i":"100","j":"150","info_format":"text/xml"};
    wms.featureinfo(queryfeature);
}
```

**Результат:** В окне браузера отобразятся метаданные объектов карты «Noginsk », отобранных в указанном местоположении.

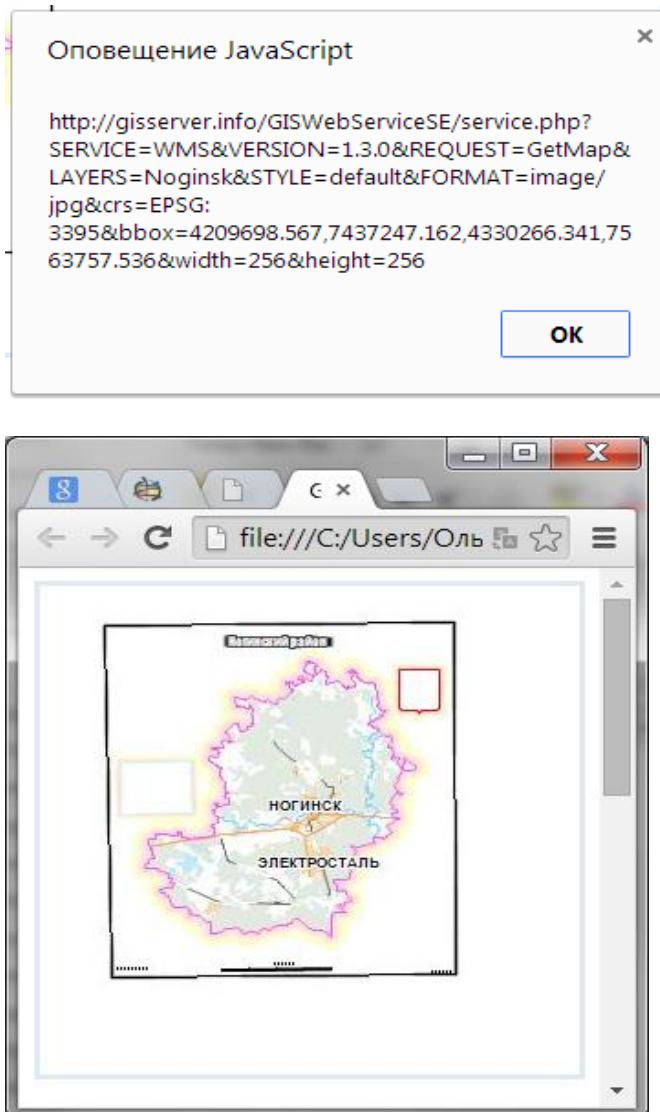


#### 6.2.3.3 Проверка создания строки запроса рисунка карты по параметрам – запрос GetMap

Для получения изображения карты необходимо создать Url запроса по параметрам при помощи метода **mapurl(querymap)** класса WmsQueries. Тестовый скрипт wmsmapUrl выводит изображение карты, полученное от сервера по сформированному запросу.

```
function wmsmapUrl() {  
  
var querymap = '{ "layers": "Noginsk", "crs": "EPSG:3395", "style": "default", "format": "image/jpg", "bbox": "4209698.567,7437247.162,4330266.341,7563757.536", "width": "256", "height": "256" }';  
  
var mapurl = wms.mapurl(querymap);  
  
alert(mapurl);  
var elem = document.getElementById("dvMap");  
if (elem == null) return;  
elem.innerHTML = "";  
var image = document.createElement('img');  
image.src = mapurl;  
elem.appendChild(image);  
}
```

**Результат:** В случае успешного выполнения в браузере отобразится всплывающее окно со строкой запроса рисунка, а затем отобразится рисунок карты – ответ сервера.



#### 6.2.4 Порядок проверки выполнения операций протокола OGC WCS

В процессе проверки выполняются запросы к сервису GIS WebService SE для получения пространственной информации о рельефе местности, а также получение описания условий получения геоданных и описания характеристик сервера по предоставлению этих данных. Запросы выполняются по протоколу OGC WCS версии 2.0.1. Проверяются операции:

- GetCapabilities,
- DescribeCoverage,
- GetCoverage.

Для выполнения запросов используются методы класса WcsQueries.

##### 6.2.4.1 Проверка получения метаданных об имеющихся на сервере геоданных покрытий и условиях получения этих данных – запрос GetCapabilities

Для получения метаданных необходимо выполнить метод **getcapabilities()** класса WcsQueries. Тестовый скрипт имеет вид:

```
function onDataLoaded(response) {  
    var elem = $get("dvMap");  
    if (elem != null) elem.innerText = response;  
}
```

```
wcs = new WcsQueries("http://gisserver.info/GISWebServiceSE/service.php");
wcs.onDataLoad = onDataLoaded;
wcs.getcapabilities();
```

**Результат:** в случае успешной проверки в браузер выводится XML-документ метаданных сервиса OGC WCS:



The screenshot shows a Microsoft Internet Explorer window displaying an XML document. The title bar reads "Home Page - My ASP.NET Application - Windows Internet Explorer". The address bar shows "http://localhost:49153/". The main content area of the browser displays the XML code for the WCS capabilities document. The XML includes details about the service provider, operations like GetCapabilities, DescribeCoverage, and GetCoverage, and various metadata elements such as coverage summary, bounding boxes, and coordinate references.

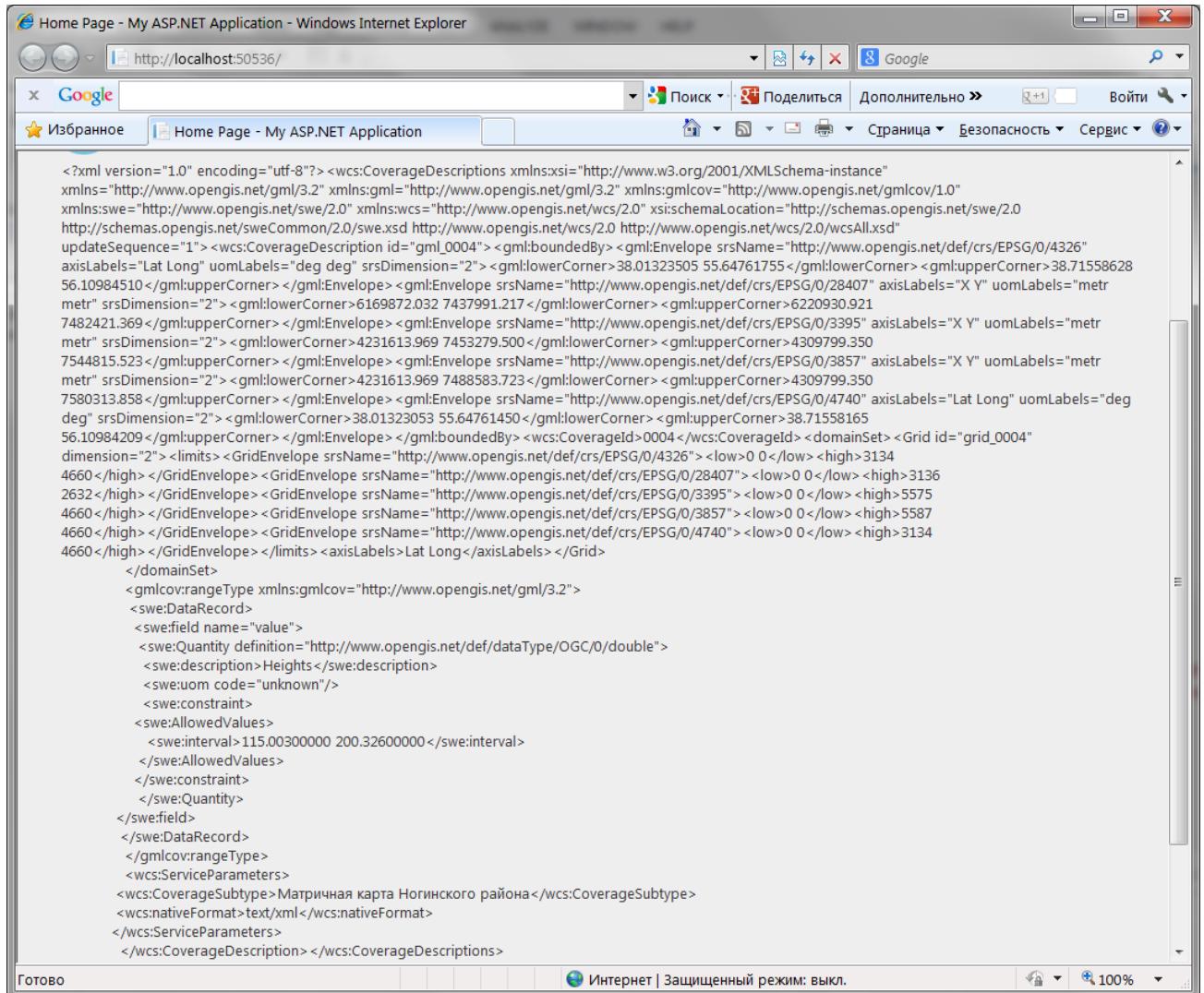
```
<ows:City>Moscow</ows:City>
<ows:AdministrativeArea>Moscow</ows:AdministrativeArea>
<ows:PostalCode>119017</ows:PostalCode>
<ows:Country>Russian Federation</ows:Country>
<ows:ElectronicEmailAddress>panorama@gisinfo.ru</ows:ElectronicEmailAddress>
</ows:Address>
</ows:ContactInfo>
</ows:ServiceContact>
</ows:ServiceProvider>
<OperationsMetadata>
<Operation name="GetCapabilities">
<DCP>
<HTTP>
<Get xlink:href="http://localhost/GISWebServiceSE/service.php?"/>
</HTTP>
</DCP>
</Operation>
<Operation name="DescribeCoverage">
<DCP>
<HTTP>
<Get xlink:href="http://localhost/GISWebServiceSE/service.php?"/>
</HTTP>
</DCP>
</Operation>
<Operation name="GetCoverage">
<DCP>
<HTTP>
<Get xlink:href="http://localhost/GISWebServiceSE/service.php?"/>
</HTTP>
</DCP>
</Operation>
</OperationsMetadata>
<wcs:ServiceMetadata>
<wcs:formatsSupported>text/xml</wcs:formatSupported>
</wcs:ServiceMetadata>
<wcs:Contents>
<wcs:CoverageSummary>
<wcs:CoverageId>0004</wcs:CoverageId>
<wcs:CoverageSubtype>Матричная карта Ногинского района</wcs:CoverageSubtype>
<wcs:Abstract>Карта территории М 1:200 000</wcs:Abstract>
<ows:WGS84BoundingBox>
<ows:LowerCorner>38.01323505 55.64761755</ows:LowerCorner>
<ows:UpperCorner>38.71558628 56.10984510</ows:UpperCorner>
</ows:WGS84BoundingBox>
</wcs:CoverageSummary>
</wcs:Contents>
</wcs:Capabilities>
```

#### 6.2.4.2 Проверка получения полного описания покрытий (3D-моделей) по списку имен покрытий – запрос DescribeCoverage

Для получения описания покрытия необходимо выполнить метод **coverageList(list)** класса WcsQueries:

```
wcs.coveragelist('0004');
```

**Результат:** в случае успешной проверки в браузер выводится XML-документ описания покрытия с именем 0004:



The screenshot shows a Microsoft Internet Explorer window with the title "Home Page - My ASP.NET Application - Windows Internet Explorer". The address bar displays "http://localhost:50536/". The main content area of the browser shows a large XML document representing the coverage description for '0004'. The XML includes various geographical coordinates and metadata fields such as 'axisLabels', 'uomLabels', and 'srsName'.

```
<?xml version="1.0" encoding="utf-8"?><wcs:CoverageDescriptions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/gml/3.2" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:gmlcov="http://www.opengis.net/gmlcov/1.0" xmlns:swe="http://www.opengis.net/swe/2.0" xmlns:wcs="http://www.opengis.net/wcs/2.0" xsi:schemaLocation="http://schemas.opengis.net/swe/2.0 http://www.opengis.net/wcs/2.0/wcsAll.xsd" http://schemas.opengis.net/wcsCommon/2.0/swe.xsd http://www.opengis.net/wcs/2.0 http://www.opengis.net/wcs/2.0/wcsAll.xsd" updateSequence="1"><wcs:CoverageDescription id="gml_0004"><gml:boundedBy><gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/4326" axisLabels="Lat Long" uomLabels="deg deg" srsDimension="2"><gml:lowerCorner>38.01323505 55.64761755</gml:lowerCorner><gml:upperCorner>38.71558628 56.10984209</gml:upperCorner></gml:Envelope><gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/28407" axisLabels="X Y" uomLabels="metr metr" srsDimension="2"><gml:lowerCorner>6169872.032 7437991.217</gml:lowerCorner><gml:upperCorner>6220930.921 7482421.369</gml:upperCorner></gml:Envelope><gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/3395" axisLabels="X Y" uomLabels="metr metr" srsDimension="2"><gml:lowerCorner>4231613.969 7453279.500</gml:lowerCorner><gml:upperCorner>4309799.350 7544815.523</gml:upperCorner></gml:Envelope><gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/3857" axisLabels="X Y" uomLabels="metr metr" srsDimension="2"><gml:lowerCorner>4231613.969 7488583.723</gml:lowerCorner><gml:upperCorner>4309799.350 7580313.858</gml:upperCorner></gml:Envelope><gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/4740" axisLabels="Lat Long" uomLabels="deg deg" srsDimension="2"><gml:lowerCorner>38.01323053 55.64761450</gml:lowerCorner><gml:upperCorner>38.71558165 56.10984209</gml:upperCorner></gml:Envelope></gml:boundedBy><wcs:CoverageId>0004</wcs:CoverageId><domainSet><Grid id="grid_0004" dimension="2"><limits><GridEnvelope srsName="http://www.opengis.net/def/crs/EPSG/0/4326"><low>0 0</low><high>3134 4660</high></GridEnvelope><GridEnvelope srsName="http://www.opengis.net/def/crs/EPSG/0/28407"><low>0 0</low><high>3136 2632</high></GridEnvelope><GridEnvelope srsName="http://www.opengis.net/def/crs/EPSG/0/3395"><low>0 0</low><high>5575 4660</high></GridEnvelope><GridEnvelope srsName="http://www.opengis.net/def/crs/EPSG/0/3857"><low>0 0</low><high>5587 4660</high></GridEnvelope><GridEnvelope srsName="http://www.opengis.net/def/crs/EPSG/0/4740"><low>0 0</low><high>3134 4660</high></GridEnvelope></limits><axisLabels>Lat Long</axisLabels></Grid></domainSet><gmlcov:rangeType xmlns:gmlcov="http://www.opengis.net/gml/3.2"><swe:DataRecord><swe:field name="value"><swe:Quantity definition="http://www.opengis.net/def/dataType/OGC/0/double"><swe:description>Heights</swe:description><swe: uom code="unknown"/><swe:constraint><swe:AllowedValues><swe:interval>115.00300000 200.32600000</swe:interval></swe:AllowedValues></swe:constraint></swe:Quantity></swe:field></swe:DataRecord></gmlcov:rangeType><wcs:ServiceParameters><wcs:CovSubtype>Матричная карта Ногинского района</wcs:CovSubtype><wcs:nativeFormat>text/xml</wcs:nativeFormat></wcs:ServiceParameters></wcs:CoverageDescription></wcs:CoverageDescriptions>
```

#### 6.2.4.3 Проверка получения данных о рельефе местности для указанного покрытия – запрос GetCoverage

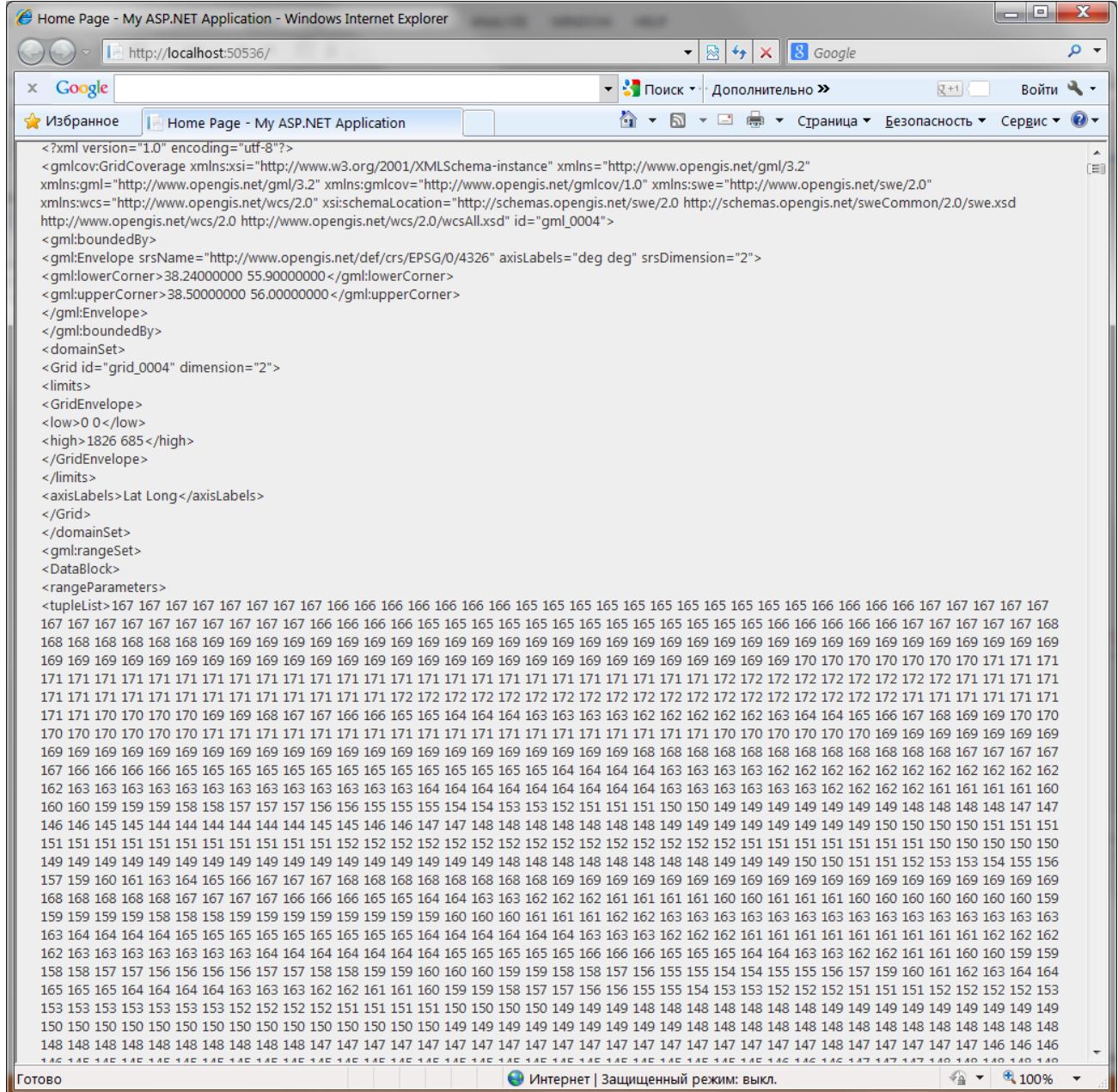
Для получения данных о рельефе местности для указанного покрытия необходимо выполнить метод **coverage(coverageoptions)** класса WcsQueries.

В тестовом примере запрашиваются данные о рельефе для покрытия 0004 с ограничением по оси X (широта) от 38.24 ° до 38.5° и ограничением по оси Y (долгота) от 55.9 ° до 56 °.

Тестовый скрипт имеет вид:

```
var coverageoptions = {'covarageid': '0004', 'dimsubsetx': 'y(38.24,38.5)', 'dimsubsety': 'x(55.9,56)'}  
wcs.coverage(coverageoptions);
```

**Результат:** в случае успешной проверки в браузер выводится XML-документ данных о рельефе местности покрытия с именем 0004 в формате GML:



The screenshot shows a Microsoft Internet Explorer window with the title "Home Page - My ASP.NET Application". The address bar displays "http://localhost:50536/". The main content area of the browser is filled with a massive XML document representing a GML file. The XML structure includes root elements like <?xml version="1.0" encoding="utf-8"?>, <gmlcov:GridCoverage>, and <gml:Envelope>. It contains numerous nested elements such as <axisLabels> Lat Long </axisLabels>, <limits>, <DataBlock>, and <rangeParameters>. The XML is extremely long, consisting of thousands of lines of text, primarily containing numerical values ranging from 160 to 171.

## 7 ПРИМЕНЕНИЕ ПРОГРАММЫ

Для работы инструментария разработки геоинформационных Web-приложений GIS WebToolKit SE необходимо установить на Web-сервере и настроить для использования картографический сервис GIS WebService SE. Картографический сервис GIS WebService SE используется картой как источник геопространственных данных.

Для подключения GIS WebToolKit SE необходимо, чтобы его компоненты были загружены в браузер вместе с кодом страницы как обычные внешние JavaScript-файлы.

### 7.1 Порядок использования

- подключить библиотеку скриптов **gwtkse.js** и файл стилей **gwtk.css** к Web-странице;
- подключить библиотеки JQuery: , jquery-1.7.1.min.js, jquery-ui-1.8.20.min.js и Modernizr-2.5.3.js;
- настроить параметры работы компонента карты;
- создать экземпляр компонента карты.

#### 7.1.1 Подключение библиотек скриптов и файла стилей

Для подключения внешних скриптов необходимо использовать элемент Script в заголовке HTML-документа.

Например,

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <script src="Scripts/gwtkse.js" type="text/javascript" ></script>
    ...
</head>
</html>,
```

где gwtkse.js – откомпилированная библиотека инструментария GIS WebToolKit SE.

Необходимо указать в заголовке:

```
<!-- подключение библиотеки GWTK -->
<script src="gwtkse.js?ver=1.3"></script>
<!-- подключение стилей GWTK -->
<link href="gwtk.css" rel="stylesheet" />
<!-- подключение библиотек jquery (необходимо для работы GWTK) -->
<script src="jquery-1.7.1.min.js" ></script>
<script src="jquery-ui-1.8.20.min.js" ></script>
<script src="modernizr-2.5.3.js" ></script>
```

#### 7.1.2 Настройка параметров работы компонента карты

Структура и назначение параметров карты и параметров слоев тайлов описаны в разделах 3.6.1 и 3.7.1 данного документа. Параметры описываются в кодировке JSON, содержат информацию для доступа к пространственным данным картографического сервиса GIS WebService SE.

Например, описание параметров демонстрационного примера **GIS WebToolKit SE** имеет вид:

```
var options = {
    "url": "http://gisserver.info/GISWebServiceSE7/wms.php", "id": "200",
    "center": [55.86, 38.349], "tilematrix": 13, "crs": 3857,
    "tilematrixset": "GoogleMapsCompatible", "count": 2,
    "layers": [{"id": "OSM", "alias": "Россия",
    "url": "http://gisserver.info/GISWebServiceSE/service.php?SERVICE=WMTS&REQUEST=GetTile&VERSION=1.0.0&LAYER=Osm&STYLE=default&TILEMATRIXSET=%tilematrixset&TILEMATRIX=%z&TILEROW=%y&TILECOL=%x&FORMAT=image%2Fpng"}]
```

```

},
{
  "id": "Noginsk", "alias": "Ногинский район", "bbox": [55.5909, 37.865, 56.1799, 38.885], "selectObject": 1,
  "url": "SER-
VICE=WMTS&REQUEST=GetTile&VERSION=1.0.0&LAYER=0002&STYLE=default&TILEMATRIXSET=%tilematrixset&TILEMATRIX
=z&TILEROW=%y&TILECOL=%x&FORMAT=image%2Fpng"
},
  "errorTileUrl": "http://gisserver.ru/gwtkse/images/empty.gif"
};

```

### 7.1.3 Создание экземпляра компонента карты

Компонент карты Мар используется для создания, удаления объекта карты и управления картой на Web-форме.

Основными параметрами карты являются область отображения карты, картографические параметры и параметры работы карты.

Область отображения карты определяется HTML-элементом для размещения рисунка карты. Параметры определяют место картографирования и выполняемые операции на карте.

Для создания карты может использоваться функция вида:

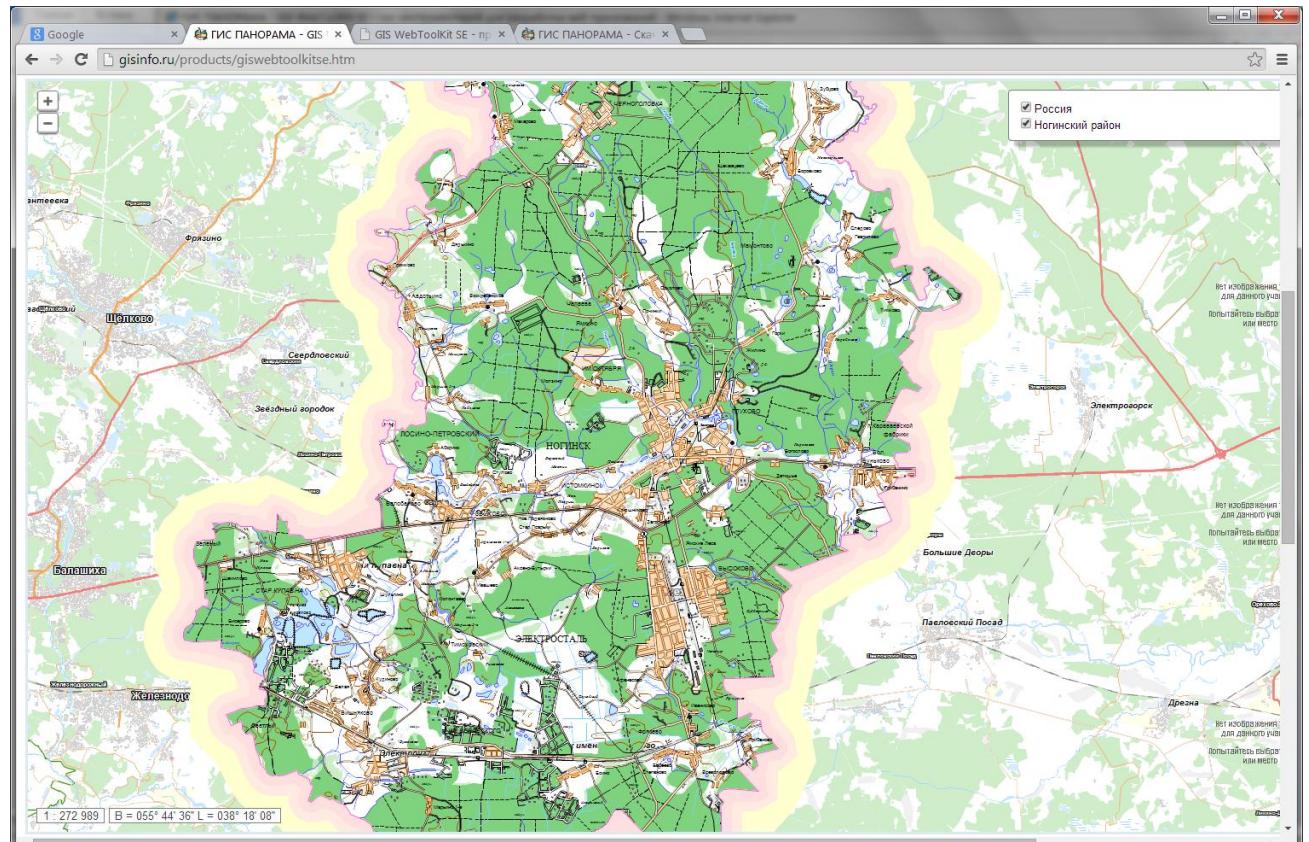
```
var map = new GWTK.Map("mapcanvas", options),
```

где mapcanvas - идентификатор HTML-элемента управления, на котором будет отображаться карта: <div id="mapcanvas" style="width:600px; height:300px;"></div>.

## 7.2 Пользовательский интерфейс

Для отображения карты пользователю достаточно ввести адрес страницы (URL), использующей библиотеку **GIS WebToolKit SE**, в Web-браузере: «<http://Web-узел/виртуальный каталог/страница>».

Например, для просмотра демонстрационного примера **GIS WebToolKit SE** необходимо ввести в Web-браузере: <http://gisinfo.ru/products/giswebtoolkitse.htm>.



Пользователю доступны операции на карте:

- перемещение изображения карты;
- изменение масштаба отображения карты;
- изменение состава отображаемых слоев карты;
- получение информации об объектах карты;
- текстовый поиск объектов.

#### 7.2.1 Перемещение изображения карты

Данная операция установлена как операция по умолчанию. Может выполняться при помощи мыши путем перетаскивания рисунка карты.

#### 7.2.2 Изменение масштаба отображения карты

Для изменения масштаба отображения карты необходимо кликнуть на кнопках «+», «-» в окне карты либо вращать колесо мыши над картой.

#### 7.2.3 Изменение состава отображаемых слоев карты

Операция позволяет получать изображения карты, различные по составу геопространственных данных.

Состав карты отображается в виде списка слоев данных карты в правом верхнем углу окна. Для управления отображением карты надо включить либо выключить кнопку элемента списка. Имя элемента списка соответствует значению алиаса слоя тайлов в параметрах слоя.

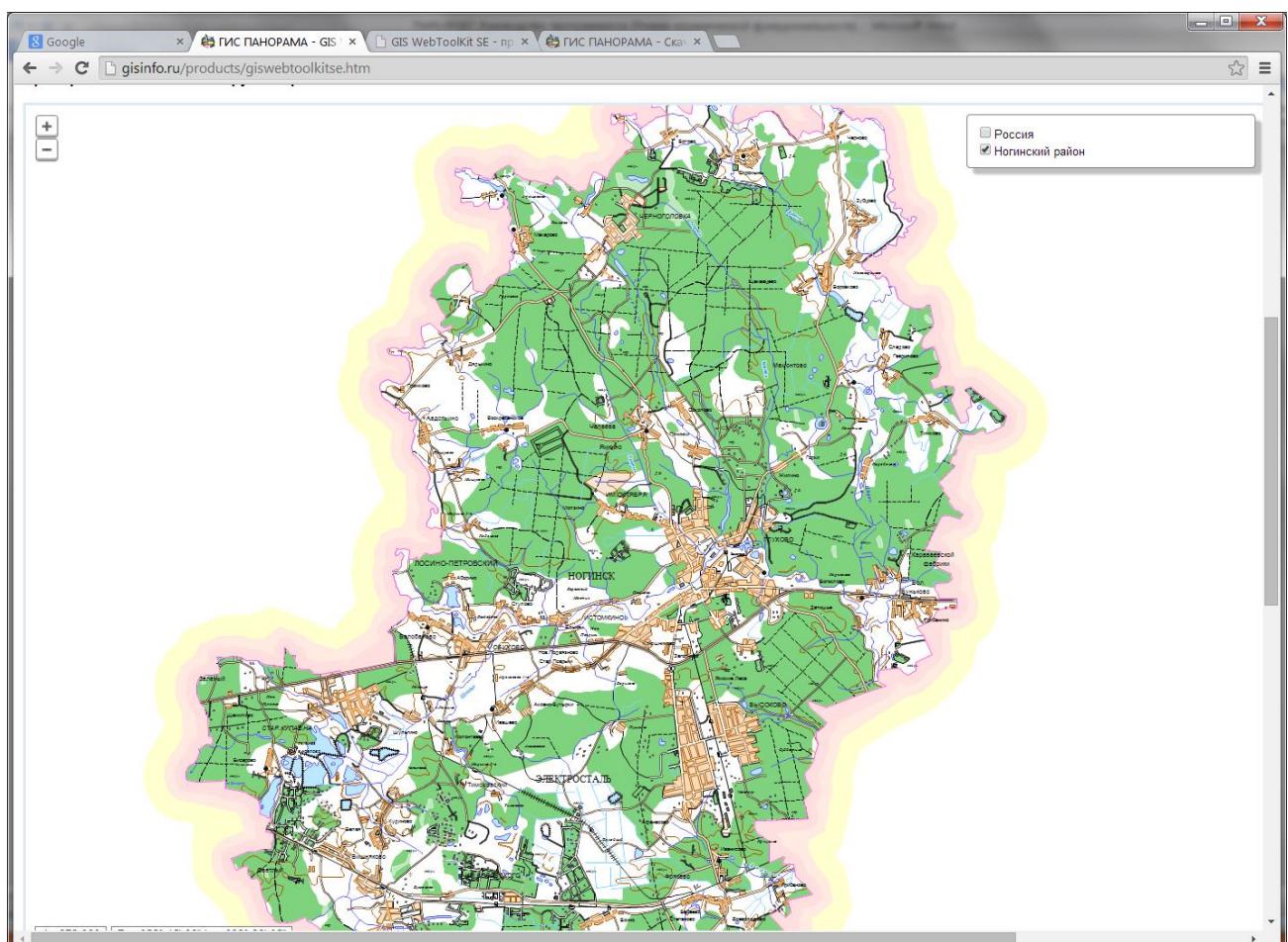


Рис. 1. Изменение состава слоев карты

#### 7.2.4 Получение информации об объектах карты

Для получения информации об объектах карты необходимо кликнуть мышью в указанном местоположении. Данные об объектах в данном местоположении выводятся на панели объектов в виде списка записей. Каждая запись содержит описание свойств отдельного объекта карты, его пространственные характеристики – площадь, периметр или длину. Маркер на карте отмечает точку выбора объектов.

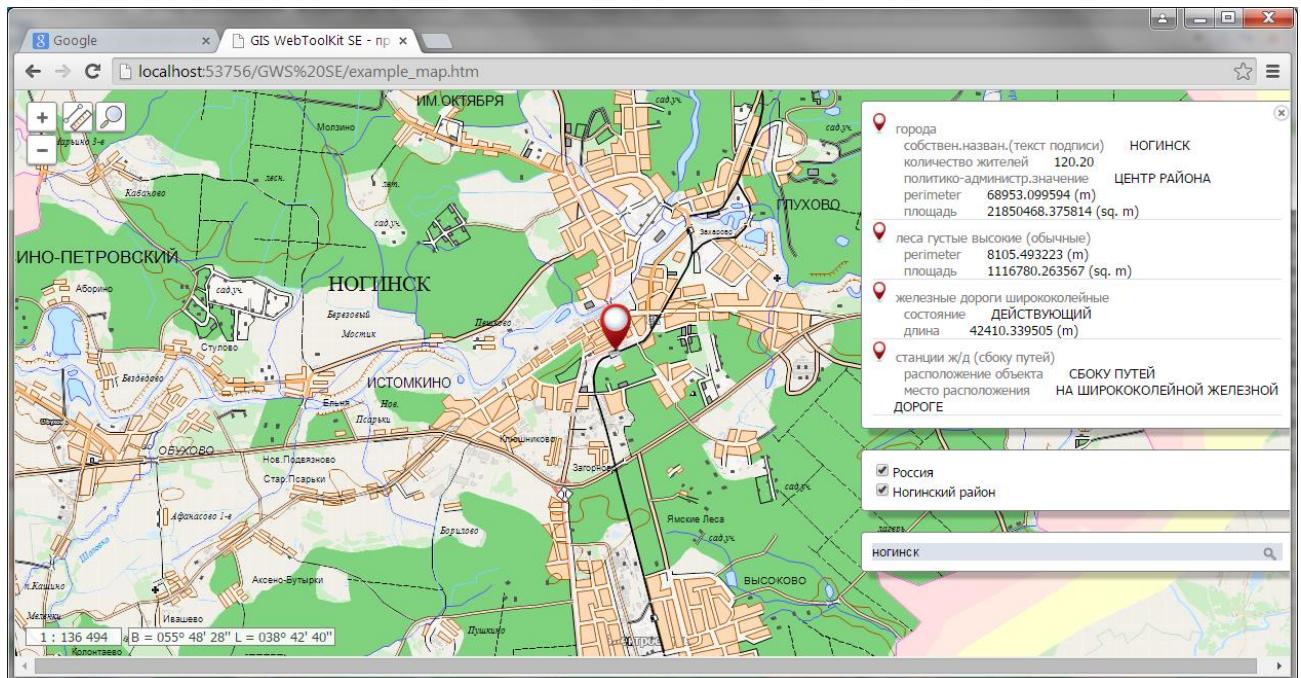


Рис. 2. Отображение свойств объектов