

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)
Кафедра автоматизации обработки информации (АОИ)

ЯЗЫК РАЗМЕТКИ ГЕОГРАФИЧЕСКОЙ ИНФОРМАЦИИ KML

Учебно-методическое пособие

Томск 2018

Жуковский О. И.

Язык разметки географической информации KML: Учебно-методическое пособие / О.И. Жуковский. – Томск, 2018. – 37 с.

© Томский государственный
университет систем управления и
радиоэлектроники, 2018

© Жуковский О.И., 2018

Оглавление

Введение.....	3
1 Разметка документов	4
2. Языки разметки SGML и XML как основа KML.....	7
3. KML и программа «Google. Планета Земля»	16
Заключение.....	37
Литература.....	37

Введение

В данном пособии рассматриваются основные понятия, методы и процедуры, связанные с разметкой документов на основе языка разметки географической информации KML.

KML (от англ. Keyhole Markup Language — язык разметки Keyhole) — язык разметки на основе XML для представления трёхмерных геопространственных данных. При описании работы с KML используется самое популярное и доступное приложение, позволяющее интерпретировать размеченные в KML сообщения – программа «Google Планета Земля» (Google Earth).

1 Разметка документов

Каждый документ, в отличие от простого гладкого текста, имеет три составляющие — содержание (смысловое наполнение), структуру и внешнее представление. Структура документа позволяет правильно определить составляющие его части и взаимоотношения между ними. Внешнее представление направлено на повышение эффективности восприятия информации читателем, что достигается за счет выделения смысловых частей документа теми или иными средствами, доступными для данной формы представления.

В документе, помимо смыслового наполнения, должна содержаться некоторая метайнформация, позволяющая определить его структуру и внешнее представление. Такая метайнформация называется *разметкой документа*.

Исторически слово разметка (markup) использовалось для описания аннотаций или других отметок в тексте, предназначенных для указания машинистке или наборщику, как именно должна быть напечатана или набрана определенная фраза. Примеры включают волнистое подчеркивание для обозначения жирного шрифта, специальные символы для обозначения пропуска отдельных предложений или их печати определенным шрифтом, и т.п. С автоматизацией форматирования и печати текстов термин был расширен, охватывая сейчас всяческие коды разметки (markup codes), вставляемые в электронные тексты для управления форматированием, печатью или иной обработкой.

Обобщая, определим *разметку или кодирование (encoding)*, как любой метод выявления интерпретации текста. На примитивном уровне все печатные тексты кодированы в этом смысле: знаки пунктуации, использование заглавных букв, размещение букв на странице, даже пробелы между словами можно считать своеобразной разметкой, функция которой - помочь читателю определить, где заканчивается одно слово и начинается

другое, или как отделить структурные элементы, такие как заголовки или элементы локальной структуры, например, подчиненные предложения.

Кодирование текста для компьютерной обработки - процесс выявления того, что неявно или предположительно, процесс указания пользователю, как интерпретировать содержимое текста.

Под **языком разметки** будем понимать набор соглашений о разметке, используемых в комплексе для кодирования текстов. Базовым средством современных технологий обработки текстовых сообщений является язык разметки XML, одна из реализаций обобщенного метаязыка разметки текстовых сообщений SGML.

Язык разметки должен специфицировать, *какая разметка является допустимой, какая - необходимой, как различаются разметка и текст сообщения, и что разметка означает.*

Разметка документа преследует следующие две основные цели:

- выделение смысловых частей (логических элементов) документа и связей между ними;
- указание действий, которые должны быть осуществлены с этими элементами.

Для достижения первой цели предназначена *структурная разметка*. Действия, направленные на получение внешнего представления, задаются *разметкой представления*.

Исторически разметка представления появилась раньше, и в течение длительного времени разметка документа была ориентирована исключительно на внешний (бумажный) вид документа. Но в последнее время ситуация существенно меняется — быстрый рост числа документов, их создание, хранение и использование в электронном виде, автоматизированная обработка и обмен документами предъявляют новые требования к разметке. В числе этих требований — независимость от среды представления, возможность осуществления эффективного поиска,

возможность повторного использования как документа целиком, так и отдельных его элементов.

Быстрый рост количества документов привел к тому, что поиск нужной информации стал занимать все больше и больше времени. Например, если необходимо найти в Интернет информацию об авторе статей по фамилии Сидоров, то простой контекстный поиск даст огромное количество ссылок на те места, где встречается данная фамилия. После чего придется либо просмотреть все полученные ссылки, либо задавать дополнительную информацию для сужения области поиска. Если бы мы могли сразу указать, что фамилию следует искать только среди авторов журнальных статей гуманитарного плана, это во много раз упростило бы поиск. Но для этого необходимо, чтобы документы, среди которых ведется поиск, были размечены должным образом с явным выделением элементов "автор", "тематика" и т.п.

Возможность повторного использования документов или отдельных его частей приводит к тому, что мы не составляем каждый раз заново отчет или деловое письмо, используем в своей работе шаблоны контрактов, изменяя лишь некоторую существенную для данного случая информацию. Но делаем мы это преимущественно вручную. Если говорить об автоматизированном формировании, связывании, повторном использовании документов, то это становится возможным только тогда, когда документы как информационные объекты являются структурированными, а используемая метаинформация полно и ясно описывает характеристики каждого элемента документа.

Все перечисленные задачи можно решить, используя исключительно структурный подход при разметке документов. Именно структурная разметка позволяет выделять смысловые элементы, определять их связи с другими элементами как в рамках одного документа, так и вне этих рамок. Далеко не всякая разметка настолько формализована, что можно говорить о языке разметки. Язык разметки должен определять ряд специальных инструкций, правил и соглашений для описания структуры элементов документа и

отношений между элементами этой структуры. Специальные инструкции, их еще называют маркерами или тэгами, в структурированных документах должны определенным образом кодироваться, то есть выделяться среди основного текста.

2. Языки разметки SGML и XML как основа KML

Стандартный обобщенный язык разметки (Standard Generalized Markup Language, SGML) был утвержден международной организацией по стандартизации (International Standards Organisation, ISO) в качестве стандарта ISO 8879:1986 в 1986 году.

SGML — это метаязык, то есть средство формального описания прикладных языков разметки, предназначенных для кодирования структурированных документов.

Разметка, определяемая в рамках SGML, основывается на двух постулатах:

- разметка должна описывать структуру документа, а не указывать, что с документом или его частями должно происходить;
- разметка должна быть строгой, чтобы программы и базы данных могли быть использованы для хранения и обработки размеченных документов.

Структура документа с точки зрения SGML представляет собой граф компонентов, вершины которого являются компонентами, а ребра — связями между ними. Основным компонентом структурированного текста является элемент. Таким образом, можно сказать, что каждый структурированный документ состоит из некоторого набора семантических элементов, связанных друг с другом по определенным правилам. Синтаксическое представление элемента документа показано на рис.1.

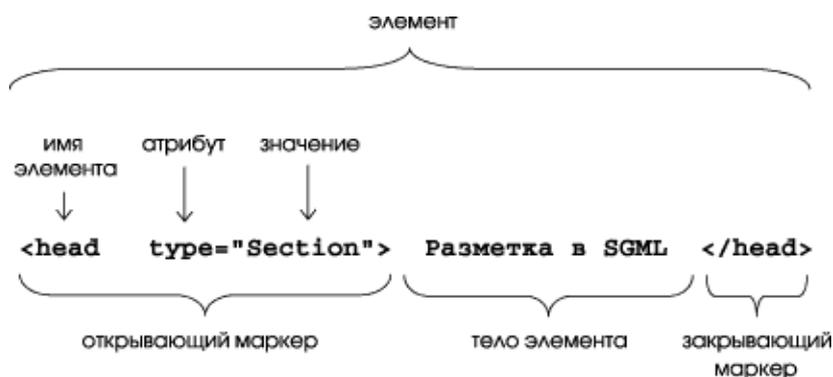


Рисунок 1 – Пример SGML-элемента.

Тело элемента (содержательный текст) обрамляется открывающим и закрывающим маркерами. Каждый маркер состоит из имени элемента, уникального для элементов одинаковой семантики, и может иметь некоторое количество атрибутов. Атрибуты предназначены для более детального описания текста среди семантически однородных элементов.

Важным достоинством SGML является то, что он не определяет заранее имена элементов и их атрибуты. Например, если автор документа считает, что семантически корректнее определить в тексте два типа списков: список фамилий и список компаний, то он может ввести два элемента `listofpeople` и `listofcompanies`. В дальнейшем эти элементы могут обрабатываться как различные семантические единицы.

Чтобы документ являлся синтаксически корректным с точки зрения SGML, необходимо, чтобы его разметка подчинялась некоторому набору правил, определяемых стандартом ISO. Одно из правил состоит в том, что допускается лишь полная вложенность одного элемента в другой. Таким образом, в каждом документе всегда будет один корневой элемент и некоторое количество иерархически вложенных элементов. (Вообще говоря, допускается наложение на документ двух независимых разметок, элементы одной из которых могут не являться вложенными в другую, но это предмет отдельного обсуждения.) Вложенность является одним из видов связей между вершинами графа компонентов.

Размеченный документ предназначен для дальнейшей обработки различными программами, каждая из которых может применять свои правила обработки к тем или иным элементам документа. Одна программа может преобразовывать текст к виду, пригодному для печати на бумаге, а другая — лишь извлекать некоторые данные (например, названия терминов) и помещать их в таблицу или базу данных.

Типы документов. Структурная разметка не предназначена для обеспечения удобочитаемости документов. Для этого существует разметка представления и соответствующие программные средства, преобразующие структурную разметку в разметку представления. Эти и другие программы, обрабатывающие документ, должны уметь распознавать элементы структуры и атрибуты элементов и применять необходимые операции к определенным элементам.

В SGML это достигается с помощью определений типов документов (Document Type Definition, DTD), посредством конструкций языка, называемых декларациями элементов. В то время как разметка документа занимается описанием семантических единиц, DTD определяет набор всех возможных разметок документов описываемого типа.

Тип документа формально определяется его составными частями и их структурой. Например, письмо можно определить как документ, имеющий реквизиты отправителя и получателя, заголовок, несколько абзацев и дату отправления. Если документ не имеет реквизитов отправителя, то, в соответствии с нашим определением, письмом он не является.

DTD определяет допустимые элементы для данного типа документа на любом из уровней вложенности, допустимое содержание каждого из элементов и набор допустимых атрибутов. При этом наличие DTD является обязательным для любого документа. Можно сказать, что в рамках SGML имеют право на существование информационные объекты, состоящие из размеченного документа и его DTD.

Декларация элементов в DTD определяет допустимое содержание как тела элемента, так и его атрибутов.

Одним из достоинств SGML является то, что он позволяет работать не только со структурированными текстами, но и с произвольными информационными объектами. Для этого вводится понятие объекта (entity).

Объектом может быть строка символов или файл (текстовый или бинарный). Для включения его в документ используется конструкция, известная в ряде языков программирования как ссылка на объект.

SGML представляет собой достаточно емкий и, в то же время, сложный метаязык. На его основе создаются языки разметки, используемые в различных областях: подготовка книг, документации, построение систем визуализации данных и т.д. Такие языки, как HTML, XML, MathML, GML, KML и многие другие созданы на основе SGML и полностью ему соответствуют.

HTML. Одна из наиболее популярных реализаций SGML язык разметки HTML родился в Лаборатории физики высоких энергий (CERN) в Женеве в 1990 году. Первоначально HTML был предназначен для разметки научных документов и их последующего совместного использования сотрудниками разных институтов и лабораторий. HTML состоял из небольшого фиксированного набора элементов — заголовков нескольких уровней, абзацев, списков и др., но главной его особенностью было использование гиперссылок и специальных меток (anchors) для указания точек перехода. Все вместе позволяло достаточно легко размечать простые документы и устанавливать связи как между ними, так и между компонентами одного документа. Человек всегда обрабатывает и анализирует информацию нелинейным образом. Поэтому возможности нелинейного хранения информации, простота использования языка разметки и широкие возможности применения привели к тому, что популярность HTML стала быстро расти и вне академических рамок. Как это часто бывает с любыми гениальными открытиями, успех превзошел все ожидания создателей.

В 1992 году HTML был формализован в качестве SGML DTD, при этом в его спецификацию была заложена возможность дальнейшего расширения. Простой синтаксис языка, в отличие от SGML, позволял создавать простые программы для анализа размеченного текста и его отображения. Начался бурный рост публикаций в HTML-формате и рост числа приложений, поддерживающих этот формат. Потребности пользователей, а также конкурентная борьба производителей программного обеспечения привели к тому, что в HTML стали добавляться неспецифицированные элементы разметки. Отсутствие строгих синтаксических правил и использование нестандартных элементов вынудили производителей программного обеспечения допускать использование синтаксически некорректных конструкций. Отметим, что в WWW найдется не так много документов, полностью удовлетворяющих общепринятым спецификациям.

За время своего существования HTML претерпел множество изменений, что весьма неприятно для создателей документов и разработчиков программ. Но гораздо большей неприятностью стало то, что, изначально задуманный как язык структурной разметки, в результате своего развития HTML превратился в язык разметки представления. Чего стоит, например, форматирование документа для улучшения его внешнего вида с помощью таблиц! Исходный текст таких документов становится практически нечитаемым, а доля полезной информации составляет лишь несколько процентов.

XML. Из предыдущего видно, что с одной стороны, максималистский подход при создании SGML привел к чрезмерной сложности языка и соответствующих программных продуктов, что неприемлемо для массового потребления. С другой стороны, простота и ограниченность HTML создавала трудности при описании сложных информационных объектов, поиске необходимой информации, создании приложений, обменивающихся данными через Интернет. Поэтому в 1996 году была сформирована рабочая группа W3C, основной задачей которой являлось создание нового языка разметки.

Этот язык должен был включать в себя гораздо больше возможностей SGML, чем HTML, но, в то же время, оставаться подходящим для использования в WWW. Чуть позже этот язык стал известен как XML (eXtensible Markup Language, расширяемый язык разметки). Разработка нового языка разметки велась около двух лет, и в начале февраля 1998 года W3C утвердила в качестве рекомендации первую спецификацию XML — XML версии 1.0.

Как и SGML, XML является метаязыком для формального описания прикладных языков разметки, предназначенных для кодирования структурированных документов. Спецификация XML определяет, как стандартным способом разметить документ, выделяя все семантически значимые компоненты.

При разработке нового языка разметки учитывались достоинства и недостатки уже существующих языков, а также то, что основным местом применения XML является Интернет. Основные требования к создаваемому языку были сформулированы следующим образом:

- XML должен быть годен к непосредственному применению в Интернет.

- XML должен быть совместимым с SGML (XML-документ должен одновременно являться и SGML-документом без внесения каких-либо изменений или дополнений).

- Число необязательных свойств в XML должно быть минимальным, в идеале нулевым (любая XML-программа должна уметь читать любой XML-документ).

- XML-документы должны быть легко читаемы с помощью простейших текстовых процессоров.

- XML-разметка должна быть простой для понимания.

Каждый размеченный с помощью XML документ начинается с пролога. Например, это инструкция `<?xml version="1.0"?>`, которая является XML-

декларацией. Ее наличие идентифицирует XML-документ и указывает, какой версии XML он соответствует.

В отличие от SGML, XML не требует обязательного определения DTD для каждого документа. При необходимости описание или указание на месторасположение DTD также помещается в прологе документа.

За прологом следует тело документа, которое представляет собой жесткую структуру элементов, подчиняющихся принципу вложенности.

Именованье элементов либо соответствует объявленному DTD, либо произвольно.

Обязательным является наличие как открывающего, так и закрывающего маркера в каждом элементе, ибо без этого при отсутствии DTD определить структуру документа невозможно.

Каждый из элементов может по аналогии с SGML содержать атрибуты, предназначенные для более детального описания семантически однородных элементов.

Возможно наличие пустых элементов, то есть элементов без содержимого. Такие элементы обозначаются с помощью символа '/' перед закрывающей угловой скобкой, например:

```
<Empty-Marker/>
```

В общем случае XML-документ может иметь шесть типов компонент:

- элементы;
- ссылки на текстовые или бинарные объекты (entity references);
- комментарии;
- инструкции обработки;
- отмеченные разделы данных (CDATA sections);
- декларация типа документов.

В данном пособии мы не будем подробно останавливаться на всех типах компонент. Отметим лишь, что инструкции обработки, в соответствии со своим названием, предназначены для предоставления информации программам, которые будут в дальнейшем обрабатывать документ. Тип

документа определяются тем же способом, что и в SGML, а отмеченные разделы данных позволяют передавать размещенные в них данные или текст "как есть", без анализа его структуры.

Что можно сказать про структурную и семантическую корректность разметки? Необязательность определения DTD, с одной стороны, существенно облегчает XML-разметку, но, с другой стороны, может значительно усложнить программы обработки. Каким образом определить в данном случае корректность XML-документа?

Чтобы определить класс правильно составленных (с точки зрения XML) документов, вводятся понятия структурной и синтаксической корректности. XML-документ является структурно корректным, если он отвечает следующим требованиям:

- Конструкция документа должна отвечать общим правилам составления документа, определенным в спецификации. В частности, некоторые конструкции (например, инструкция `<?xml version="1.0"?>`) могут присутствовать только в определенных местах документа.
- Никакой атрибут не используется более одного раза в одном маркере элемента.
- Значения атрибутов не ссылаются на внешние объекты.
- Все непустые элементы удовлетворяют принципу вложенности.
- Все используемые объекты продекларированы.
- Нет ссылок на бинарные объекты непосредственно из текста. Такие ссылки возможны лишь в момент декларации объекта.
- Текстовые объекты не являются рекурсивными.

По определению, если документ не является структурно корректным, то он не является и XML-документом.

При наличии у документа DTD возможна его проверка на синтаксическую корректность. При этом XML-документ считается синтаксически корректным, если он, во-первых, является структурно

корректным, а, во-вторых, полностью соответствует всем правилам, изложенным в соответствующем DTD.

3. KML и программа «Google. Планета Земля»

Наверняка многие из вас пользовались уникальной программой “Google. Планета Земля” (Google Earth) (<http://www.google.com/earth/index.html>) или хотя бы слышали о ней. Эта программа позволяет на экране компьютера совершать “виртуальные путешествия” по всей нашей планете, просматривая спутниковые аэро- и фотоснимки, рельефные карты, отдельные здания в 3D, знакомиться с интересными географическими материалами, а также добавлять собственные пометки с фотографиями и описаниями различных достопримечательностей и просматривать такие пометки, оставленные другими пользователями. Данная программа сегодня очень популярна, количество ее скачиваний, по заявлению компании Google, уже превысило сотни миллионов, и даже если в это число входят неоднократные скачивания разных версий программы одним и тем же пользователем, все равно нужно признать, что количество ее активных пользователей поистине колоссально.

Заметим, что проект “Google. Планета Земля” — это далеко не единственный web-ресурс, предоставляющий доступ к космическим фотоснимкам. Желающие, например, могут посмотреть их на сайте <http://kosmosnimki.ru>, причем по качеству отечественные спутниковые фотоснимки практически не уступают американским в сопоставимом разрешении. Но почему большинство пользователей, даже в России, все же предпочитает работать с продуктом компании Google?

Возможно, ответ кроется в том, что специалисты Google не просто предложили для поиска и просмотра космические фотоснимки, а превратили их в совершенно новый программный продукт. Иными словами, космические снимки и соответствующие им карты стали основой для своеобразного браузера, функционирующего, как и обычные web-браузеры, на базе специального языка разметки для описания пространственных объектов KML.

Язык KML (аббревиатура фразы Keyhole Markup Language — “язык разметки Keyhole”) — это специализированный язык разметки, созданный на основе языка XML для представления трехмерных геопространственных данных и являющийся сегодня международным стандартом Open Geospatial Consortium. Этот язык был разработан компанией Keyhole Inc., которая была приобретена компанией Google в 2004 году. (Название “Keyhole” — это дань уважения спутникам космической разведки серии KeyHole, обеспечивающим американцев фотоснимками начиная с 1976 года.). Программа же “Google. Планета Земля” стала первой программой, использующей этот язык, — хотя она уже не является единственной.

Как уже было сказано, язык KML создан на основе стандарта XML, а тот, в свою очередь, в качестве прототипа использовал языки SGML и HTML.

Однако в отличие от языка HTML в языке XML и всех других языках разметки, “произшедших” от XML, в записи тегов учитывается регистр, поэтому их необходимо вводить в точности так, как указано в справочном руководстве.

Создавать KML-файлы можно с помощью программы “Google. Планета Земля”, обладающей графическим интерфейсом специально для этой цели. А можно воспользоваться простым XML-редактором и вводить KML-код с нуля. Файлы с разметкой на языке KML (так же, как файл HTML) можно создать в любом текстовом редакторе (например, в стандартном приложении “Блокнот”). Если вы будете создавать его самостоятельно, то не забудьте сохранить текст в кодировке UTF-8, иначе русские буквы в программе “Google. Планета Земля” будут отображаться некорректно. Расширение имени файла, как нетрудно понять, должно быть kml.

KML-файлы модели и относящиеся к ним изображения можно упаковать в KMZ-архив, так чтобы весь материал находился в одном контейнере. Если вы хотите поделиться своими KML или KMZ-файлами, то можете присоединить их к электронному письму в виде вложения, поместить в общую папку в домашней или корпоративной сети или выложить на веб-

сервер. После того как вы правильно сконфигурируете веб-сервер и опубликуете адрес своего KML-файла, любой человек, установивший Google Earth (или какое-нибудь совместимое приложение), сможет созданный вами файл просмотреть.

Можно без труда персонализировать свои KML-презентации, воспользовавшись стилями для описания значков, всплывающих информационных окон, цвета, линий, форм и надписей. KML позволяет также отображать определенные детали в зависимости от момента времени в заданном интервале и изменять отображение в соответствии с выбранным пользователем масштабом, увеличивая уровень детализации, когда зритель приближает изображение.

Простые KML-документы

Самые простые KML-документы можно создавать прямо в интерфейсе Google Планеты Земля; для этого даже не потребуется текстовый редактор. Таким способом можно создавать и изменять метки, наложения на земную поверхность, пути и многоугольники.

Метки. Метка является одним из самых часто используемых компонентов в Google Планете Земля. Она служит для указания положения на земной поверхности и представлена значком булавки желтого цвета. В самом простом варианте метка содержит только элемент <Point>, который определяет ее положение. Вы также можете указать ее название и персонализированный значок или добавить другие элементы.

Откройте файл KML_Samples.kml в Google Планете Земля и перейдите в папку Placemarks. В ней представлены три типа меток: *простая* (*simple placemarks*), *плавающая* (*floating placemark*) и *выдавленная* (*extruded placemark*) (рис. 2).

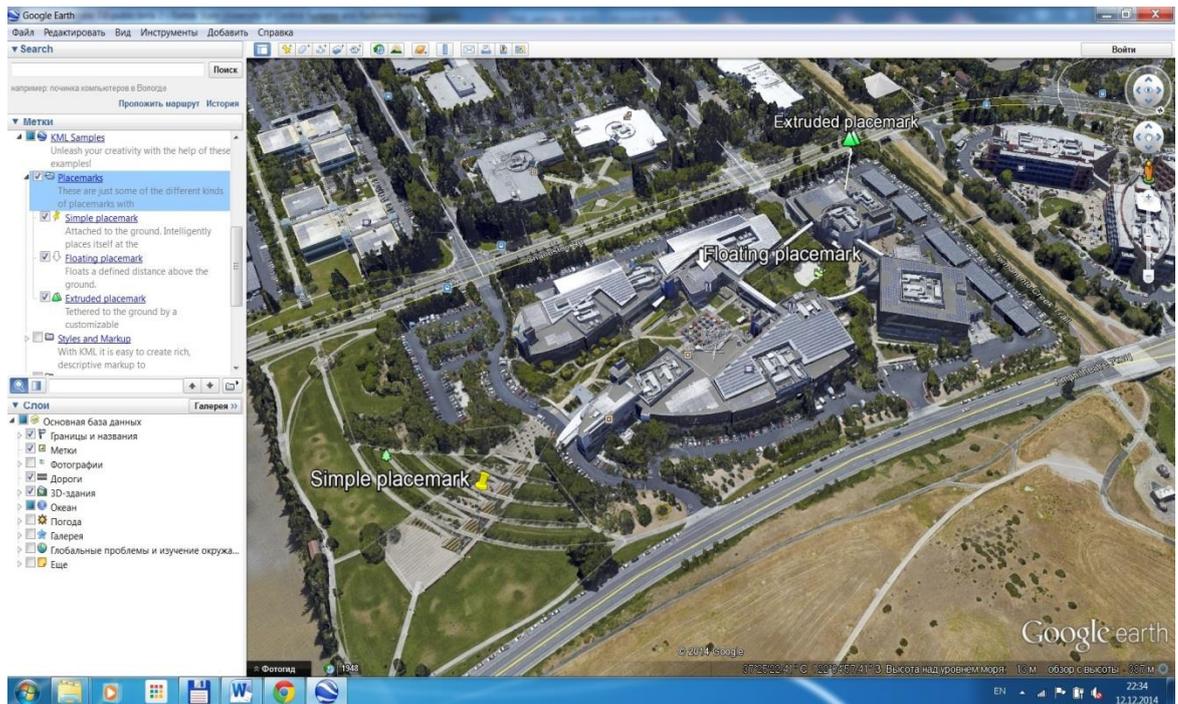


Рисунок 2 – Окно Google Earth после открытия файла KML_Samples.kml

Разметка простой метки на языке KML выглядит следующим образом.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Placemark>
<name>Простая метка</name>
<description> Привязана к земной поверхности. Приспосабливается к
рельефу местности.</description>
<Point>
<coordinates>-122.0822035425683,37.42228990140251,0</coordinates>
</Point>
</Placemark> </kml>
```

Рассмотрим структуру этого файла.

Заголовок XML: с него начинается каждый KML-файл. Перед заголовком не должно быть никаких символов или пробелов.

Декларация пространства имен KML: вторая строка каждого файла формата KML 2.2.

Объект Placemark, содержащий следующие элементы:

- *Name* – имя, которое используется в качестве ярлыка метки;
- *Description* – описание, которое отображается во всплывающем окне, привязанном к метке;
- *Point* – координаты, определяющие положение метки на земной поверхности (*долгота и широта*, иногда также *высота*).

Если вы не можете найти эту метку на карте, подсказываем: она расположена прямо на 41-м корпусе Google – именно там разрабатывалась Google Планета Земля.

То, что пользователи Google Планеты Земля видят как метку, является элементом <Placemark> с дочерним элементом <Point> в формате KML. Это единственный способ отобразить метку с ярлыком в окне 3D-просмотра. По умолчанию метка имеет вид уже знакомой вам желтой булавки. В KML-коде элемент <Placemark> может содержать один или несколько базовых векторных элементов, таких как ломаные линии (LineString), многоугольники (Polygon) или модели (Model), но только <Placemark> с дочерним элементом <Point> может отображаться как метка с ярлыком. Элемент <Point> служит для правильного размещения метки на карте, но сам не имеет графического представления.

Описательный HTML в метках. В файле KML_Samples.kml приведены исчерпывающие примеры форматирования текста метки, включая добавление ссылок, изменение размера и стиля шрифта, выравнивание и использование таблиц. Чтобы просмотреть полный список возможностей, скопируйте и вставьте в текстовый редактор пример Descriptive HTML (в папке Styles and Markup). Для выделения кликните правой кнопкой мыши на имени данной папки и выберите операцию копирования.

Этот файл (так же, как файл HTML) можно создать в любом текстовом редакторе (например, в стандартном приложении “Блокнот”). Если вы будете создавать его самостоятельно, то не забудьте сохранить текст в формате Unicode, иначе русские буквы в программе “Google. Планета Земля” будут отображаться некорректно. Расширение имени файла, как нетрудно понять, должно быть kml.

Авторазметка в Google Планете Земля 4.0 и более поздних версий. В Google Планете Земля предусмотрена функция авторазметки, которая преобразует текст наподобие *www.google.com* в активные гиперссылки. Текст, заключенный в теги <description> или <Snippet>, а также в элемент <text> внутри <BalloonStyle>, автоматически преобразуется в стандартные HTTP-ссылки. Добавлять теги вручную не нужно.

Использование элемента CDATA. Чтобы добавить стандартный HTML-код в элемент <description>, его необходимо заключить в тег CDATA. Если этого не сделать, угловые скобки придется записывать в виде ссылок на объекты, иначе HTML-код будет неправильно анализироваться (например, символ >потребуется писать как **>**, а символ < – как **<**). Это особенность языка XML в целом, а не только Google Планеты Земля.

Обратите внимание на HTML-разметку с тегами CDATA и без них. Сначала рассмотрим элемент <description> с тегами CDATA.

Пример 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Placemark>
      <name>Пример с тегами CDATA</name>
      <description>
        <![CDATA[ <h1>Теги CDATA имеют смысл!</h1>
```

<p>Без ссылок на объекты текст <i>удобнее читать</i> и
проще писать.

</p>]]>

</description>

<Point>

<coordinates>102.595626,14.996729</coordinates>

</Point>

</Placemark>

</Document>

</kml>

Обратите внимание, что при проведении экспериментов с данным примером его после редактирования сохранять нужно как файл с кодировкой UTF-8, иначе надписи на русском языке будут выглядеть не корректно.

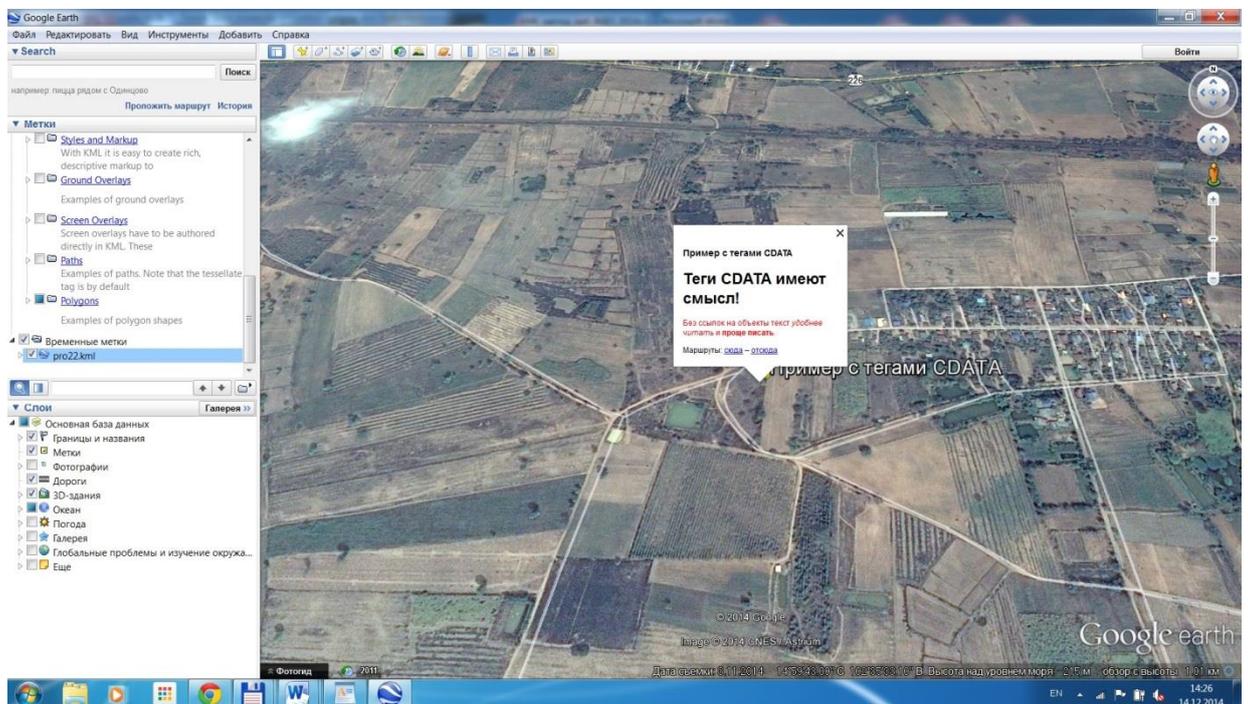


Рисунок 3 – Окно Google Earth после открытия файла с примером 1

Теперь посмотрим, как оформляется <description> без тегов CDATA.

Пример 2.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
<Placemark>
<name>Пример ссылок на объекты</name>
<description> &lt;h1&gt;Ссылки на объекты так неудобно
записывать!&lt;/h1&gt; &lt;p&gt;&lt;font color="green"&gt;Без ссылок на
объекты текст &lt;i&gt;удобнее читать&lt;/i&gt; и &lt;b&gt;проще
писать&lt;/b&gt;.&lt;/font&gt;&lt;/p&gt;
</description>
<Point>
<coordinates>102.594411,14.998518</coordinates>
</Point>
</Placemark>
</Document>
</kml>
```

Наложения на земную поверхность позволяют "накрыть" рельеф местности определенным изображением. Элемент `<Icon>` содержит ссылку на *JPG*-файл изображения. Ниже приведен пример из файла *KML_Samples.kml*, показывающий извержение вулкана Этна в 2001 году.

Пример 3.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Folder>
<name>Наложения на земную поверхность</name>
<description>Примеры наложений</description>
<GroundOverlay>
<name>Крупное наложение</name>
```

```

<description>Извержение Этны 13 июля 2001 года.</description>
<Icon>
<href>http://developers.google.com/kml/documentation/images/etna.jpg</href>
</Icon>
<LatLonBox>
<north>37.91904192681665</north>
<south>37.46543388598137</south>
<east>15.35832653742206</east>
<west>14.60128369746704</west>
<rotation>-0.1556640799496235</rotation>
</LatLonBox>
</GroundOverlay>
</Folder>
</kml>

```

Обратите внимание на то, что первые две строки файла точно такие же, как в первом примере: XML-заголовок и декларация пространства имен KML.

Для группировки и обозначения содержания в примере используется папка Ground Overlays. Обратите внимание на то, что она появляется в панели "Метки", как только вы загружаете файл KML_Samples.kml в Google Планету Земля.

Расположение наложения определяется с помощью элемента <LatLonBox>, в котором указываются крайние значения южной и северной широты, а также восточной и западной долготы. Кроме того, если ось у изображения не направлена строго на север, задается значение поворота. В примере используется JPEG-изображение, но Google Планета Земля также поддерживает графические форматы BMP, GIF, TIFF, TGA и PNG.

Пути. В Google Планете Земля можно создавать пути разных типов, что позволяет подойти к отображению данных творчески. В KML-разметке путь представлен элементом <LineString>. Изучите пример Absolute Extruded в папке Paths: линия определена с помощью указанного в примере приведенного ниже кода.

Пример 4.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
<name>Пути</name>
<description>Примеры путей. Значение тега tessellate по умолчанию
равно 0. Чтобы создать ломаную линию, значение необходимо
изменить непосредственно в KML-коде
.</description>
<Style id="yellowLineGreenPoly">
<LineStyle>
<color>7f00ffff</color>
<width>4</width>
</LineStyle>
<PolyStyle>
<color>7f00ff00</color>
</PolyStyle>
</Style> <Placemark>
<name>Линия вдоль земной поверхности</name>
<description>Прозрачная зеленая стена с желтыми краями</description>
<styleUrl>#yellowLineGreenPoly</styleUrl>
<LineString>
<extrude>1</extrude>
<tessellate>1</tessellate>
<altitudeMode>absolute</altitudeMode>
```

```

<coordinates> -112.2550785337791,36.07954952145647,2357
-112.2549277039738,36.08117083492122,2357
-112.2552505069063,36.08260761307279,2357
-112.2564540158376,36.08395660588506,2357
-112.2580238976449,36.08511401044813,2357
-112.2595218489022,36.08584355239394,2357
-112.2608216347552,36.08612634548589,2357
-112.262073428656,36.08626019085147,2357
-112.2633204928495,36.08621519860091,2357
-112.2644963846444,36.08627897945274,2357
-112.2656969554589,36.08649599090644,2357
</coordinates>
</LineString>
</Placemark>
</Document>
</kml>

```

Обратите внимание на то, что это одна и та же линия, отображаемая на определенной высоте над землей. Благодаря элементу `<tessellate>` линия разбивается на короткие отрезки, а элемент `<extrude>` отвечает за ее выравнивание относительно земной поверхности.

Многоугольники. С помощью многоугольников можно моделировать здания и другие объекты. Примеры можно найти в папке Polygons файла KML_Samples.kml.

Модель здания Пентагона построена так: внутренний и внешний контуры "вытянуты" вниз к земной поверхности. Ниже показан пример разметки.

Пример 5.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<kml xmlns="http://www.opengis.net/kml/2.2">
<Placemark>
  <name>Пентагон</name>
  <Polygon>
    <extrude>1</extrude>
    <altitudeMode>relativeToGround</altitudeMode>
    <outerBoundaryIs>
      <LinearRing>
        <coordinates>
          -77.05788457660967,38.87253259892824,100
          -77.05465973756702,38.87291016281703,100
          -77.05315536854791,38.87053267794386,100
          -77.05552622493516,38.868757801256,100
          -77.05844056290393,38.86996206506943,100
          -77.05788457660967,38.87253259892824,100
        </coordinates>
      </LinearRing>
    </outerBoundaryIs>
    <innerBoundaryIs>
      <LinearRing>
        <coordinates>
          -77.05668055019126,38.87154239798456,100
          -77.05542625960818,38.87167890344077,100
          -77.05485125901024,38.87076535397792,100
          -77.05577677433152,38.87008686581446,100
          -77.05691162017543,38.87054446963351,100
          -77.05668055019126,38.87154239798456,100
        </coordinates>
      </LinearRing>
    </innerBoundaryIs>
  </Polygon>
</Placemark>
</kml>
```

</Polygon>

</Placemark>

</kml>

Для векторизации границ полигонов (многоугольников) в программе Google Earth необходимо выбрать инструмент «Добавить многоугольник». Затем, в появившемся диалоге «Google Планета Земля – Создать: Многоугольник» (рис. 4) указать: название, описание и стиль многоугольника. Удобнее выбрать стиль линии красного цвета контра и область без заливки – Контур. Далее, не закрывая диалог, необходимо указать точки контура, последовательно щелкая мышкой по изображению его границ.

Процесс создания каждого нового контура начинается с выбора инструмента «Добавить многоугольник», а заканчивается нажатием кнопки «ОК» в диалоге «Google Планета Земля – Создать: Многоугольник» (рис. 4).

В процессе векторизации не забывайте указывать название объекта, представленного многоугольником, и его описание. Если внутри контура многоугольника имеются участки не занятые данным многоугольником, их тоже необходимо векторизовать.

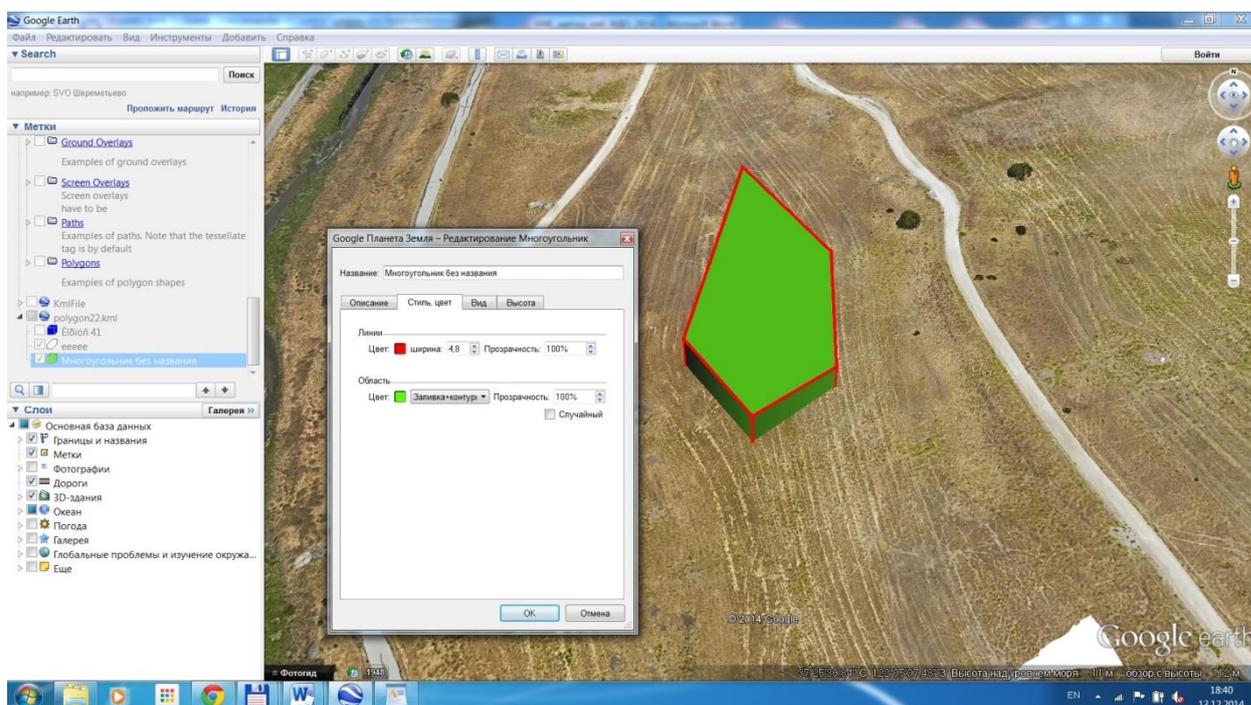


Рисунок 4 – Пример диалога с Google Earth в процессе создания полигона.

Расширенные KML-документы

В этом разделе представлены некоторые элементы KML, которые можно задать только с помощью текстового редактора, такие как общие стили геометрии, выделенные значки меток и наложения на экране. Написание KML-кода вручную требует больше знаний и опыта, чем создание компонентов в интерфейсе Google Планеты Земля, однако немного попрактиковавшись, вы сможете без труда добавлять эти эффекты в KML-файлы.

Стили геометрии. Вы уже знаете, как создавать компоненты в интерфейсе Google Планеты Земля и как выглядит их KML-разметка. Наверняка вы обратили внимание на то, что стили играют важную роль при отображении данных. Со временем вам может потребоваться определять собственные стили.

Если определить элемент `<Style>` в начале KML-документа и присвоить ему идентификатор, этот стиль можно будет применять к геометрическим

элементам, меткам и наложениям, определенным в других разделах документа. Так как один и тот же стиль можно применять к нескольким элементам, используется термин *общие стили*. Достаточно один раз определить `<Style>`, чтобы затем сослаться на него с помощью элемента `<styleUrl>`. Если элемент `<Style>` определен в том же файле, перед его идентификатором необходимо добавить знак решетки (#). Если же он определен в другом файле, в элементе `<styleUrl>` необходимо указать полный URL.

В файле `KML_Samples.kml` представлено несколько общих стилей, которые определены в начале файла и обозначены идентификаторами. Рекомендуем использовать описательные идентификаторы – так вам будет проще отличать стили друг от друга. Ниже показан пример стиля `transBluePoly`, который определяет для многоугольников прозрачную голубую заливку и белые границы толщиной 1,5 (белый цвет задан по умолчанию). Этот стиль использован в модели 41-го корпуса Google (в папке `Polygons`).

Пример 6.

```
<?xml version="1.0" encoding="UTF-8"?>

<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Style id="transBluePoly">
      <LineStyle>
        <width>1.5</width>
      </LineStyle>
      <PolyStyle>
        <color>7dff0000</color>
      </PolyStyle>
    </Style>
    <Placemark>
```

```
<name>Корпус 41</name>
<styleUrl>#transBluePoly</styleUrl>
<Polygon>
<extrude>1</extrude>
<altitudeMode>relativeToGround</altitudeMode>
<outerBoundaryIs>
<LinearRing>
<coordinates> -122.0857412771483,37.42227033155257,17
-122.0858169768481,37.42231408832346,17
-122.085852582875,37.42230337469744,17
-122.0858799945639,37.42225686138789,17
-122.0858860101409,37.4222311076138,17
-122.0858069157288,37.42220250173855,17
-122.0858379542653,37.42214027058678,17
-122.0856732640519,37.42208690214408,17
-122.0856022926407,37.42214885429042,17
-122.0855902778436,37.422128290487,17
-122.0855841672237,37.42208171967246,17
-122.0854852065741,37.42210455874995,17
-122.0855067264352,37.42214267949824,17
-122.0854430712915,37.42212783846172,17
-122.0850990714904,37.42251282407603,17
-122.0856769818632,37.42281815323651,17
-122.0860162273783,37.42244918858722,17
-122.0857260327004,37.42229239604253,17
-122.0857412771483,37.42227033155257,17
</coordinates>
</LinearRing>
</outerBoundaryIs>
</Polygon>
```

</Placemark>

</Document>

</kml>

Обратите внимание на то, что <Style> является дочерним элементом элемента <Placemark>, а не геометрического элемента, к которому он применяется.

Стили выделенных значков. Пример Highlighted Icon в папке Styles and Markup демонстрирует, как задать эффект выделения в KML. В документе необходимо определить два стиля: normalPlacemark (для метки в обычном состоянии) и highlightPlacemark (для выделения метки, на которую наведен указатель мыши). В элементе <StyleMap> содержатся две пары "ключ-значение", определяющие стиль значка в зависимости от его состояния. Состояния может быть два: *обычное* и *выделенное* (при наведении мыши).

Чтобы определить стиль значка при наведении мыши, выполните указанные ниже действия.

1. Создайте элемент <Style> для значка метки в обычном состоянии и присвойте ему идентификатор (в примере это normalPlacemark).
<Style> содержит элемент <Icon> со ссылкой <href>, указывающей на используемое изображение, как показано в примере ниже.
2. Создайте элемент <Style> для значка метки в выделенном состоянии и присвойте ему идентификатор (в примере это highlightPlacemark).
3. Создайте элемент <StyleMap> и присвойте ему идентификатор (в примере это exampleStyleMap), на который будет ссылаться компонент Placemark.
4. Определите внутри элемента <StyleMap> ссылку на стиль #normalPlacemark для метки в обычном состоянии (*normal*).

5. Определите внутри элемента <StyleMap> ссылку на стиль #highlightPlacemark для выделенной метки (*highlight*).
6. Добавьте к компоненту Placemark элемент <styleUrl>, ссылающийся на "#exampleStyleMap".

Пример 7.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Выделенный значок</name>
    <description>Наведите указатель мыши на значок, чтобы увидеть, как он
выглядит в выделенном состоянии</description>
    <Style id="highlightPlacemark">
      <IconStyle>
        <Icon>
          <href>http://maps.google.com/mapfiles/kml/paddle/red-stars.png</href>
        </Icon>
      </IconStyle>
    </Style>
    <Style id="normalPlacemark">
      <IconStyle>
        <Icon>
          <href>http://maps.google.com/mapfiles/kml/paddle/wht-blank.png</href>
        </Icon>
      </IconStyle>
    </Style>
    <StyleMap id="exampleStyleMap">
      <Pair>
        <key>normal</key>
```

```

<styleUrl>#normalPlacemark</styleUrl>
</Pair>
<Pair>
<key>highlight</key>
<styleUrl>#highlightPlacemark</styleUrl>
</Pair>
</StyleMap>
<Placemark>
<name>Наведите указатель мыши на значок</name>
<styleUrl>#exampleStyleMap</styleUrl>
<Point>
<coordinates>-122.0856545755255,37.42243077405461,0</coordinates>
</Point>
</Placemark>
</Document>
</kml>

```

Наложения на экране. В отличие от наложений на земную поверхность, наложения на экране нельзя создавать в интерфейсе Google Планеты Земля. Примеры можно найти в папке Screen Overlays файла KML_Samples.kml.

В качестве иллюстрации включите папку Absolute Positioning: Top left файла KML_Samples.kml. В левом верхнем углу окна просмотра появится наложение. Ниже в примере показана его KML-разметка.

Пример 8

```

<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<ScreenOverlay>
<name>Absolute Positioning: Top left</name>

```

```

<Icon>
<href>http://developers.google.com/kml/documentation/images/top_left.jpg
</href>
</Icon>
<overlayXY x="0" y="1" xunits="fraction" yunits="fraction"/>
<screenXY x="0" y="1" xunits="fraction" yunits="fraction"/>
<rotationXY x="0" y="0" xunits="fraction" yunits="fraction"/>
<size x="0" y="0" xunits="fraction" yunits="fraction"/>
</ScreenOverlay>
</kml>

```

Размещение наложения определяется следующим образом: точка изображения, заданная элементом `<overlayXY>`, привязывается к точке на экране, заданной элементом `<screenXY>`. В данном случае левый верхний угол изображения (0,1) выравнивается по левому верхнему углу экрана.

Просмотрите другие примеры в папке, чтобы лучше понять принцип определения фиксированных изображений и изображений, размер которых меняется в зависимости от размера экрана. Обратите внимание на то, что для большей точности вместо *xunits* и *yunits* можно использовать пиксели.

Сетевые ссылки. Сетевая ссылка состоит из элемента `<Link>` с гипертекстовой ссылкой `<href>`, указывающей на загружаемый файл. Элемент `<href>` может содержать локальную ссылку или абсолютный URL. Таким образом, `<NetworkLink>` может служить не только для загрузки файлов из сети.

Элемент `<href>` может указывать на расположение любого из следующих файлов:

- файла изображения, который используется в стилях значков, наложениях на земную поверхность или наложениях на экране;
- файла модели, которая используется в элементе `<Model>`;

- KML- или KMZ-файла, загружаемого с помощью сетевой ссылки.

Такой файл может располагаться локально или на удаленном сервере. В самом простом варианте сетевые ссылки помогают разделить крупный KML-файл на несколько меньших файлов на одном и том же компьютере, чтобы с ними было удобнее работать.

До сих пор мы рассматривали только случаи, в которых KML-код передается в Google Планету Земля с локального компьютера. Сетевые ссылки позволяют получать содержание с удаленных серверов и, как правило, используются для предоставления данных большому количеству пользователей. Таким образом, если в содержание необходимо внести изменения, достаточно сделать это в файле источника, и все пользователи автоматически увидят обновленную версию.

Заключение

После знакомства с методическим руководством в обязательном порядке следует посмотреть ресурс <https://developers.google.com/kml/?hl=ru>, являющийся самым полным русскоязычным сводом стандартов и практических примеров разметки геопространственных объектов на языке KML.

Самый лучший способ изучить KML - экспериментировать с имеющимися файлами, меняя разные значения и наблюдая за результатами в геобраузере «Google Планета Земля» (Google Earth). Если браузер ничего не показывает, значит, вы где-то допустили ошибку. В программе Google Earth есть механизм проверки ошибок, который, возможно, будет вам полезен. (Выберите из меню пункт Инструменты => Настройки и на вкладке Общие в разделе Обработка ошибок KML отметьте переключатель Показывать сообщения об ошибках.) Кроме того, для проверки своего KML-кода можете воспользоваться какой-нибудь специализированной программой. Одна такая программа, разработанная компанией Galdos Systems, имеется на сайте www.kmlvalidator.com.

Литература

1. Джози Вернеке Язык географической разметки KML. - М.: ДМК Пресс, 2010. - 288
2. Руководство по KML
https://developers.google.com/kml/documentation/kml_tut?hl=ru