

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ**

**Кафедра автоматизации обработки информации**

**О. И. Жуковский**

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В УПРАВЛЕНИИ**

**Методические указания  
по выполнению лабораторных работ  
для студентов направления подготовки 230102  
«Государственное и муниципальное управление»**

**Томск  
2017**

## СОДЕРЖАНИЕ

Введение.....	3
1 Общая характеристика самостоятельной работы по дисциплине .....	4
2 Выполнение лабораторных работ по дисциплине .....	4
2.1 Лабораторная работа «Разработка функциональной модели процесса создания хранилища данных».....	5
2.2 Разметка географической информации на языке KML.....	19
3 Оформление отчетов о выполнении лабораторных работ.....	41
Приложение А .....	43
Варианты заданий к лабораторной работе .....	43
«Разработка функциональной модели процесса создания хранилища данных» .....	43
Приложение Б .....	45
Варианты заданий к лабораторной работе .....	45
«Разметка географической информации на языке KML».....	45
Приложение В.....	47
Форма титульного листа.....	47

## ВВЕДЕНИЕ

«Информационные технологии в управлении» относится к вариативной части основных дисциплин блока I (Дисциплины (модули)) по направлению подготовки 230102 «Государственное и муниципальное управление» (уровень бакалавриата). Знания, полученные в процессе ее изучения, в дальнейшем потребуются при выполнении учебно-исследовательских и научных работ, а также при выполнении выпускной квалификационной работы.

Целью данной дисциплины является формирование и расширение у студентов фундамента современной информационной культуры; обеспечение навыков работы на персональном компьютере в условиях принятия управленческих решений; получение навыков использования типовых программных пакетов обработки информации; развитие способности в условиях постоянного роста достижений науки и техники приобретать новые знания, используя современные информационные технологии.

Задачами дисциплины является изучение основных теоретических вопросов и рассмотрение существующего российского и зарубежного практического опыта по выбору, внедрению и применению информационных систем в организации, знакомство с принципами и стандартами построения корпоративных информационных систем, их классификацией и программной структурой.

Данные методические указания предназначены для организации самостоятельной работы и выполнения лабораторных работ по разделам «Информационные системы обработки данных», «CASE – технологии» и «Геоинформационная технология» дисциплины «Информационные технологии в управлении».

Лабораторные работы выполняются в соответствии с порядком, описанным в методических указаниях.

Выбор варианта лабораторной работы осуществляется по общим правилам с использованием следующей формулы:

$$V = (N \times K) \operatorname{div} 100,$$

где  $V$  – искомый номер варианта,

$N$  – общее количество вариантов,

$\operatorname{div}$  – целочисленное деление,

при  $V = 0$  выбирается максимальный вариант,

$K$  – код варианта.

## **1 ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ**

Самостоятельная работа является важной составляющей в изучении дисциплины и состоит из следующих видов деятельности: самостоятельное изучение теоретического материала, выполнение лабораторной работы и написание отчета о выполнении лабораторной работы.

Самостоятельная работа над теоретическим материалом направлена на изучение основных понятий дисциплины. К этой деятельности относится:

- изучение тем теоретического курса в соответствии с учебной программой дисциплины;
- проработка вопросов для самопроверки в конце глав учебного пособия;
- подготовка и выполнение лабораторной работы.

## **2 ВЫПОЛНЕНИЕ ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ**

*Лабораторная работа* – одна из основных форм организации учебного процесса. При выполнении лабораторных работ обучающиеся овладевают первоначальными профессиональными знаниями и навыками, приобретают опыт творческой и исследовательской деятельности, овладевают современными методами практической работы с применением техниче-

ских средств и программного обеспечения, развивают профессиональные компетенции.

## **2.1 Лабораторная работа «Разработка функциональной модели процесса создания хранилища данных»**

*Цели и задачи работы* Целью данной работы является получение и развитие навыков в планировании бизнес-процессов разработки хранилища данных со стороны заказчика на основе функциональных моделей, отвечающих методологии IDEF0.

Рассматриваются основные бизнес-процессы разработки хранилища данных.

Для самостоятельного моделирования предлагаются варианты предметных областей, предполагающих необходимость использования хранилищ данных в ходе процессов принятия управленческих решений.

Трудоемкость работы – 6 часов.

В ходе выполнения лабораторной работы необходимо построить функциональную модель бизнес-процессов разработки хранилища данных в одной из предметных областей, предполагающих необходимость использования хранилищ данных в ходе процессов принятия решений при их управлении. При рассмотрении предметной области необходимо выявить не менее трех источников оперативных данных, которые послужат основой для загрузки проектируемого хранилища данных и определяют класс задач, решаемых на основе проектируемого хранилища данных. Например, для управления недвижимым имуществом это будут БТИ (Ростехинвентаризация), земельный комитет и Учреждение юстиции по регистрации прав на недвижимое имущество. Для аптечной сети — центральный склад и несколько аптек и т. п.

Выполнение задания рекомендуется проводить согласно следующим этапам.

*Сбор информации.* Наиболее подходящая стратегия выполнения данного этапа — провести поиск информации об особенностях управления выбранным процессом в сети Интернет. Если у Вас есть знакомые, имеющие отношения к подобному процессу, то опросите их для получения более полного ощущения специфики данного этапа.

Предложите список лиц и документов, которые, на ваш взгляд могли бы служить источниками информации при моделировании реального процесса разработки ХД в данной области.

*Начало моделирования.* Создайте диаграммы А0 и А-0. Обратите внимание, что эти две диаграммы должны полностью рассказывать все о моделируемом процессе с минимальной степенью детализации. Диаграмма А-0, часто называемая контекстной диаграммой, определяет все необходимые связи моделируемого процесса с окружающим миром.

В первую очередь создайте диаграмму А0 и, обобщив ее, создайте диаграмму А-0.

Особое внимание обратите на то, что каждая модель должна иметь определенную цель и создаваться с конкретной точки зрения. Выбор цели осуществляется с учетом вопросов, на которые должна ответить модель, а выбор точки зрения — в соответствии с выбором позиции, с которой описывается система. Не забывайте, что автор данной модели должен стоять на точке зрения управленца, планирующего использовать информацию, собираемую в хранилище, для обеспечения качества принимаемых управленческих решений.

Построенные вами на данном этапе диаграммы А-0 и А0 должны представлять законченную картину процесса создания ХД в выбранной области деятельности, поскольку они отражают все основные входы, управления, выходы и функции моделируемого процесса.

*Продолжение моделирования.* Продолжение моделирования основывается на тех же методах, что и начальный этап, и выводит модель на следующий уровень детализации. Создайте отдельную диаграмму для, возможно, каждого блока диаграммы верхнего уровня, затем постройте для всех блоков новые диаграммы, и так до тех пор, пока модель не будет описывать объект с нужной для достижения вашей цели степенью детализации.

Помните, что в результате моделирования Вы должны сформировать достаточный для разработки ХД набор документов и определить механизмы их создания.

*Подготовка отчета.* По завершении моделирования подготовьте отчет в формате doc или pdf, содержащий качественное описание выбранной Вами области бизнеса с указанием использованных источником информации и набор диаграмм IDEF0, представляющий функциональную модель процесса разработки Хранилища Данных для выбранной вами области бизнеса и содержащую основные бизнес-процессы разработки хранилища данных, представленные в данных методических указаниях. В случае использования программ, автоматизирующих построение функциональной модели, файлы моделей, полученные данными программами, в отчете не предоставляются.

Модель может быть построена как с применением программ, автоматизирующих процесс построения моделей IDEF0 (Design/IDEF, BPWin, ERWin), так и просто любых программ, позволяющих нарисовать диаграммы по правилам IDEF0, например, Microsoft Visio.

После знакомства с методическим руководством рекомендуется посмотреть ресурс [www.vernikov.ru](http://www.vernikov.ru), являющийся одним из самых полных русскоязычных сводов стандартов по семейству IDEF-технологий (в случае отсутствия времени на доскональное изучение данного ресурса в пол-

ном объеме можно сразу перейти к разделу <http://vernikov.ru/biznes-modelirovanie/tehnologii-i-standarty.html>).

*Задание для выполнения лабораторной работы.* Согласно варианта задания необходимо взять из Приложения А одну из предметных областей, предполагающих необходимость использования хранилищ данных в ходе процессов принятия решений при их управлении.

### ***Теоретические сведения***

Рассмотрим типовую модель бизнес-процессов разработки ХД, которая может быть положена в основу реализации любого конкретного проекта. Эта модель содержит минимально достаточное число обязательных этапов для реализации небольшого или среднего по масштабу проекта.

Проект разработки ХД начинается после того, как выбраны инструменты разработки и сформирована команда проекта. В проектный цикл разработки ХД обычно включаются следующие типовые процессы (этапы):

- формулирование требований;
- создание вычислительной среды;
- моделирование данных;
- определение процедур извлечения преобразования и загрузки данных;
- проектирование аналитических отчетов;
- разработка приложений ХД;
- настройка производительности;
- проверка качества;
- передача системы складирования данных в эксплуатацию.

Каждый типовой этап разработки ХД будем описывать по следующей схеме:



- **Описание задачи.** Что обычно должно быть достигнуто в течение данного этапа разработки ХД.
- **Временные требования.** Приблизительная оценка количества времени для решения задачи данного этапа.
- **Результат выполнения этапа.** В конце каждого этапа оформляются один или более документов, которые описывают шаги и результаты решения данной задачи.
- **Потенциальные опасности.**

Теперь перейдем к рассмотрению отдельных компонентов представленной бизнес-модели.

### **Формулирование требований**

**Задача этапа.** Главной задачей данного этапа является идентификация требований будущих пользователей ХД и оформление их в виде документа «Каталог требований». Обычно сбор требований осуществляется путем опроса групп потенциальных пользователей ХД на специальных совещаниях и переговорах. Конечные пользователи, как правило, не знакомы с концепцией ХД и процессом складирования данных. Поэтому для успешного решения этой задачи важна помощь лиц, принимающих решения (ЛПР), т.е. руководителей организации. На этом этапе определяются:

- масштаб проекта создания ХД (границы предметной области);
- перечень и содержание отчетов;
- требования по анализу данных (перечень задач анализа данных);
- требования к аппаратному обеспечению;
- требования к системному программному обеспечению;
- значения базовых параметров Хранилища данных (скорость обработки запросов, объемы используемых данных, актуализация данных, производительность системы и т. д.);
- требования к квалификации персонала (программа обучения персонала);

перечень источников данных;

конкретизация плана реализации проекта разработки ХД (определяется дата завершения проекта).

В дополнение, на основе собранной выше информации составляется план восстановления хранилища данных в случае аварийных сбоев. Разрабатывается стратегия архивирования и восстановления ХД.

**Временные требования.** Время выполнения этапа — от двух недель до двух месяцев.

**Результатом выполнения этапа** являются каталог требований, утвержденный заказчиком, и уточненный план проекта, который точно определяет используемые ресурсы и даты контрольных точек проверки хода выполнения проекта.

**Потенциальные опасности.** Этап формулирования требований часто оказывается одним из самых узких мест проекта создания ХД. Причина состоит в конфликте внутрикорпоративных интересов и в необходимости наладить коммуникации для успешного выполнения и этапа, и проекта в целом.

По определению складирование данных предполагает интеграцию в ХД данных из нескольких источников (подразделений организации). Позиция и взгляды подразделений на бизнес-информацию (производство данных, их верификацию, поставку данных в ХД, разграничение доступа к данным и т. д.) могут сильно расходиться и даже быть диаметрально противоположными. Даже если команда разработчиков предложит блестящее решение по созданию ХД, его реализация может споткнуться о нежелание определенных групп потенциальных пользователей предоставлять данные в ХД или конструктивно участвовать в определении требований к системе складирования данных.

Если не удастся наладить коммуникации между участниками процесса, то все усилия по созданию ХД будут потрачены впустую и проект

никогда не будет завершен в установленные сроки, а в худшем случае просто будет провален.

Одним из способов избежать такой ловушки является непосредственное вовлечение руководителей организации в процесс реализации проекта. Следует заручиться поддержкой влиятельного покровителя из числа высшего руководства, чтобы можно было влиять на позицию подразделений сотрудничать с командой разработчиков ХД.

### **Создание вычислительной среды для хранилища данных**

**Задача этапа.** Главной задачей этого этапа является создание информационно-вычислительной среды, в которой будет разрабатываться ХД. Это достаточно важный этап. Вычислительная среда разработки ХД данных должна достаточно точно моделировать ту вычислительную среду, в которой реально будет эксплуатироваться ХД, т. е. она должна быть масштабируемой по аппаратному решению.

После того как требования к программно-аппаратной части определены, нужно обеспечить установку серверов приложений, серверов БД, клиентских рабочих станций и автоматизированных мест разработчиков.

На этом этапе должна быть достигнута полная ясность в том, как будут выполняться технологические работы по созданию ХД.

Причина, по которой лучше применять отдельную вычислительную среду для разработки, состоит в том, что внесение изменений и тестирование разрабатываемого ХД могут привести к аварийным сбоям в существующей вычислительной среде организации, что повлечет дополнительные эксплуатационные издержки.

### **Временные требования.**

Закупка и установка серверов — до двух недель.

Монтажные работы на установке компьютерной сети — две-четыре недели.

**Результатом выполнения этапа** являются спецификации на программно-аппаратное обеспечение, а также скрипты и установки для программного обеспечения.

**Потенциальные опасности.** Самой большой опасностью является использование одного сервера БД для моделирования различных вычислительных сред, например, вычислительной среды разработки и вычислительной среды тестирования, или, что еще хуже, для вычислительной среды разработки и вычислительной среды эксплуатации ХД, особенно если на этом сервере работает уже существующая информационная система.

В этом случае могут возникнуть конфликты между различными участниками проекта создания ХД и ИТ-службой организации в случае непредусмотренной остановки сервера БД.

Кроме того, в тестовой среде будет невозможно промоделировать ожидаемые показатели нагрузки на систему и оценить ее производительность.

### **Моделирование данных**

**Задача этапа.** Главной задачей этапа является разработка логической и физической моделей данных для ХД. Этот этап — один из самых важных для проекта создания ХД. Ошибки и просчеты, допущенные на этом этапе, будет очень сложно исправить на последующих этапах. Кроме того, подобные просчеты могут привести к пересмотру всего проекта и, следовательно, к его фактическому провалу.

При выполнении этого этапа сначала строится логическая модель данных, которая впоследствии преобразуется в физическую модель.

Одной из подзадач этого этапа являются идентификация и описание источников данных, которые также могут стать подзадачей этапа определения процедур извлечения, преобразования и загрузки данных в ХД (ETL-процессов).

**Временные требования.** Время выполнения этого этапа — от двух недель до двух месяцев.

**Результатом выполнения этапа** являются перечень источников данных и их описание, а также логическая и физическая модели данных.

**Потенциальные опасности.** Самой большой опасностью при выполнении этого этапа является самоуверенность проектировщиков ХД. Во многих случаях даже опытные проектировщики допускают от двух до пяти ошибок в структуре данных (потерь существенных семантических зависимостей в данных) на проект. Причиной таких ошибок часто становится недостаточная осведомленность проектировщиков о предметной области ХД и низкое качество информации, поставляемой аналитиками предметной области.

Бизнес-аналитики предметной области могут не предоставить полной информации о функциональных зависимостях в данных, что приведет в результате к проектированию частично неправильной схемы данных.

Хорошей предупредительной мерой для предотвращения подобных ситуаций является привлечение квалифицированных экспертов, особенно в случае если проект разработки ХД выполняется силами самой организации.

### **Определение процедур извлечения, преобразования и загрузки данных**

**Задача этапа.** Главной задачей этапа является идентификация и определение процедур извлечения, очистки (фильтрации), преобразования и загрузки данных. Это весьма трудоемкий этап, не столько по временным затратам, сколько по усилиям, предпринимаемым по анализу структур данных источников и подающих систем.

Исключительно редким является случай, когда ХД данных создается на голом месте, т.е. в подразделениях отсутствуют автоматизированные подсистемы обработки данных. Как правило, данные уже существуют (в

том или ином виде). Их нужно собрать, согласовать, привести к единому формату, агрегировать и загрузить в ХД. По этой причине этот этап является характерным для проекта создания ХД.

Также следует помнить о том, что сам процесс подготовки и загрузки данных в создаваемое ХД может занять более половины времени, отведенного на реализацию проекта, особенно в проектах большого масштаба.

**Временные требования.** Время выполнения этапа — от одной недели до полутора месяцев.

**Результатом выполнения этапа** являются схема соответствия данных подающих систем и ХД, программы или ETL-инструменты.

**Потенциальные опасности.** Первой потенциальной опасностью при выполнении этого этапа является недооценка временных параметров. Обычно на выполнение этапа выделяют немного времени. Процедуры извлечения, преобразования и загрузки данных в ХД оказывают непосредственное влияние на качество и полноту предоставляемой информации конечным пользователям. Недостаточное внимание к разработке этих процедур может вызвать негативные реакции у пользователей после получения ими отчетов.

Второй потенциальной опасностью является стремление команды разработчиков сделать процесс ETL как можно более всеобъемлющим, мотивируя свои действия стремлением обеспечить качество данных. Следует помнить, что главная цель ETL-процесса — оптимизации скорости загрузки данных в ХД.

### **Проектирование аналитических отчетов**

**Задача этапа.** Главной задачей выполнения этого этапа является проектирование и разработка аналитических отчетов на спроектированной структуре данных. Это также специфичный этап для проекта ХД.

Перечень требуемых отчетов содержится в «Каталоге требований», и их разработка решает в целом поставленную задачу. Однако следует пом-

нить, что потенциальные пользователи не всегда точно знают, что они хотят увидеть в отчетах. С другой стороны, как правило, собранные данные предоставляют большие возможности по формированию разнообразных отчетов, чем это зафиксировано в «Каталоге требований». Здесь должен быть найден разумный компромисс.

**Временные требования.** Время выполнения этого этапа зависит от числа разрабатываемых отчетов. В зависимости от сложности отчета его разработка занимает от 4 часов до двух недель.

**Результатом выполнения этапа** станут спецификация кубов данных (измерения и метрики) и разработанные отчеты.

**Потенциальные опасности.** Потенциальной опасностью при выполнении этого этапа является то, что не уделяется достаточного внимания оптимизации времени получения отчета. Конечный пользователь не любит долго ждать. А если он получал такой отчет на своей настольной системе быстрее, то мнение о ХД у него будет, мягко говоря, неадекватное.

Хороший способ избежать такой опасности — тестирование разработанных отчетов с целью минимизации времени их получения.

### **Разработка приложений хранилища данных**

**Задача этапа.** Главной задачей данного этапа является формирование программной среды, в которой пользователи будут извлекать данные из ХД и просматривать predetermined отчеты.

Каковы бы ни были инструментальные средства OLAP, необходимо продумать, как будут происходить визуализация отчетов и их доставка конечному пользователю.

**Временные требования.** Срок выполнения этого этапа — от одной недели до месяца. При использовании тонкого клиента типа браузера следует исходить из того, что в среднем на разработку и отладку одной веб-страницы тратится полдня.

**Результатом выполнения этапа** будет документация, описывающая механизм доставки пользователям отчетов и спецификации экранных форм.

**Потенциальные опасности.** Самой большой потенциальной опасностью является ложное представление о достаточной квалификации пользователей ХД для работы с ИТ-технологиями. Конечные пользователи хотят быстро получать данные, необходимые для решения своих конкретных задач, а не изучать многотомные инструкции по работе с программным обеспечением.

Если интерфейс интуитивно не понятен или не содержит ясных и кратких инструкций по работе с электронной формой (встроенных в форму Help), он не будет использоваться системой складирования данных с должной эффективностью, что приведет к постепенному ее вымиранию.

### **Настройка производительности**

**Задача этапа.** Главная задача выполнения этого этапа — добиться оптимальной производительности ETL-процессов, производства отчетов и их доставки конечному пользователю.

Извлечение, преобразование и загрузка данных, как правило, является длительным процессом, который выполняется в пакетном режиме с не очень высоким приоритетом. Пока данные не загружены в ХД, новые отчеты не могут быть получены.

При обработке некоторых запросов захватывается большое количество данных, на которых выполняются динамические операции агрегирования и суммирования, что приводит к значительному времени производства отчета.

На доставку отчетов пользователю может влиять загрузка локальной вычислительной сети и большой объем сетевого трафика.

**Временные требования.** Время выполнения этого этапа не должно превышать одну-две недели.



**Результатом выполнения этапа** является перечень рекомендаций по настройке производительности.

**Потенциальные опасности.** Потенциальной опасностью этого этапа считается использование вычислительной среды разработки ХД, которая не масштабируется к вычислительной среде эксплуатации ХД. Из-за этого рекомендации по настройке производительности не будут соответствовать реальным условиям эксплуатации ХД.

### **Проверка качества**

**Задача этапа.** Главная задача этапа — убедиться, что ХД готово к эксплуатации. Как правило, проверка качества выполняется отдельной группой специалистов, не входящих в состав команды разработчиков.

**Временные требования.** Срок выполнения этого этапа — от одной до четырех недель.

**Результатом выполнения этапа** являются план тестирования ХД и заключение о готовности ХД к эксплуатации.

**Потенциальные опасности.** Самый большой риск любого проекта — это люди: их квалификация, амбиции, заинтересованность в деле, мотивы и т. д. Поскольку люди, проверяющие ХД на этом этапе, не входят в проектную команду, возникает потенциальная опасность, связанная с недостаточной образованностью этих людей в области складирования данных. Разумным решением при выполнении этого этапа является привлечение организацией сторонних специалистов высокой квалификации по проверке качества ХД.

### **Передача системы ведения хранилища данных в эксплуатацию**

**Задача этапа.** Главная задача этапа — передача системы ведения хранилища данных заказчику и представление ее конечным пользователям.

**Временные требования.** Срок выполнения этого этапа — от одного дня до нескольких недель.

**Результатом выполнения этапа** является акт о приемке-сдаче программного продукта.

**Потенциальные опасности.** Самая большая потенциальная опасность для этого этапа — неготовность потенциальных пользователей к работе с ХД. Хорошим правилом здесь будет проведение нескольких презентаций и коротких обучающих курсов по работе с ХД.

### **Сопровождение и модификация хранилища данных**

Обычно в проектный цикл включают еще два этапа — этап сопровождения ХД и этап его модификации. Это важные этапы в жизни ХД. Однако, как показывает опыт, целесообразно выделять их в самостоятельные проекты.

Было бы большой ошибкой поручать разработчикам ХД его сопровождение. Процессы сопровождения ХД требуют от ИТ-специалистов иной квалификации, чем процессы его разработки. Чтобы сопровождать, не обязательно уметь писать программы, но обязательно нужно уметь их настраивать и использовать.

Если необходимость в модернизации ХД возникает спустя несколько месяцев после сдачи его в эксплуатацию, это говорит о том, что проект не был успешным. Потребность в модернизации реально может сформироваться спустя шесть месяцев после интенсивной его эксплуатации, когда проверены его возможности, прочувствована отдача и видны новые перспективы использования, т.е. когда ХД стало частью производственного процесса. При этом не факт, что та же команда разработчиков будет заниматься его модернизацией.

### **Резюме**

Процесс разработки ХД может быть представлен в виде модели бизнес-процессов. Бизнес-модель процесса разработки позволяет:

отобразить субъективное мнение руководителя автоматизируемой организации и ключевых участников команды разработчиков на процесс проектирования конкретного ХД;

учесть особенности ИТ-проекта, в рамках которого проектируется ХД;

достаточно быстро составить план проектирования конкретного ХД;

просчитать длительность проектных работ (создать временную модель проектирования).

Представленная в настоящем пособии укрупненная бизнес-модель процесса разработки ХД является достаточно общей. В нее не включены многие детали и риски. Конкретизация и детализация процесса разработки ХД — это основная задача и проекта разработки ХД. Руководитель автоматизируемой организации должен понимать, какие документы он вправе ожидать от членов команды разработчиков, какие документы должен предоставить и кому разработанные документы должен передать.

## **2.2 Разметка географической информации на языке KML**

*Цели и задачи работы:* ознакомление с основными принципами разметки географической информации на языке KML в рамках векторной модели представления пространственных данных.

Трудоемкость работы – 6 часов.

*Задание для выполнения лабораторной работы.* Согласно варианта задания Приложения Б провести разметку на основе языка KML одного из маршрутов, приведенных в перечне.

В ходе выполнения контрольной работы вы должны будете разметить один из вариантов маршрута. В маршруте должны быть представлены не менее 11 точек. Для каждой точки создайте метку с презентацией дан-

ного местоположения с использованием разметки на HTML. Метки соседних точек должны иметь разные стили. В начале, конце и по ходу маршрута на основе полигонов (многоугольники) покажите не менее 5 зданий, сооружений или участков территории с учетом их высоты. Учтите, что разные части здания могут иметь разную высоту. Для представления зданий используйте разные стили. Сам маршрут должен быть представлен линейным объектом и явно выделяться на фоне растровой подложки.

Полученный в результате kml-файл представить в качестве отчета.

*Задачи*, решаемые при выполнении лабораторной работы:

- Получение навыков использования языков разметки;
- Получение навыков работы с географическими объектами в рамках векторной модели пространственных данных;
- Знакомство с географическими сервисами сети Интернет на примере Google Earth.;

После знакомства с методическим руководством в обязательном порядке следует просмотреть 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 програм-

мой. Одна такая программа, разработанная компанией Galdos Systems, имеется на сайте [www.kmlvalidator.com](http://www.kmlvalidator.com).

Основные конструкции языка разметки KML приведены в прилагающемся к пособию файле `KML Samples.kml`.

### *Теоретические сведения*

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, “произошедших” 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 стан-

дартном 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. 1).

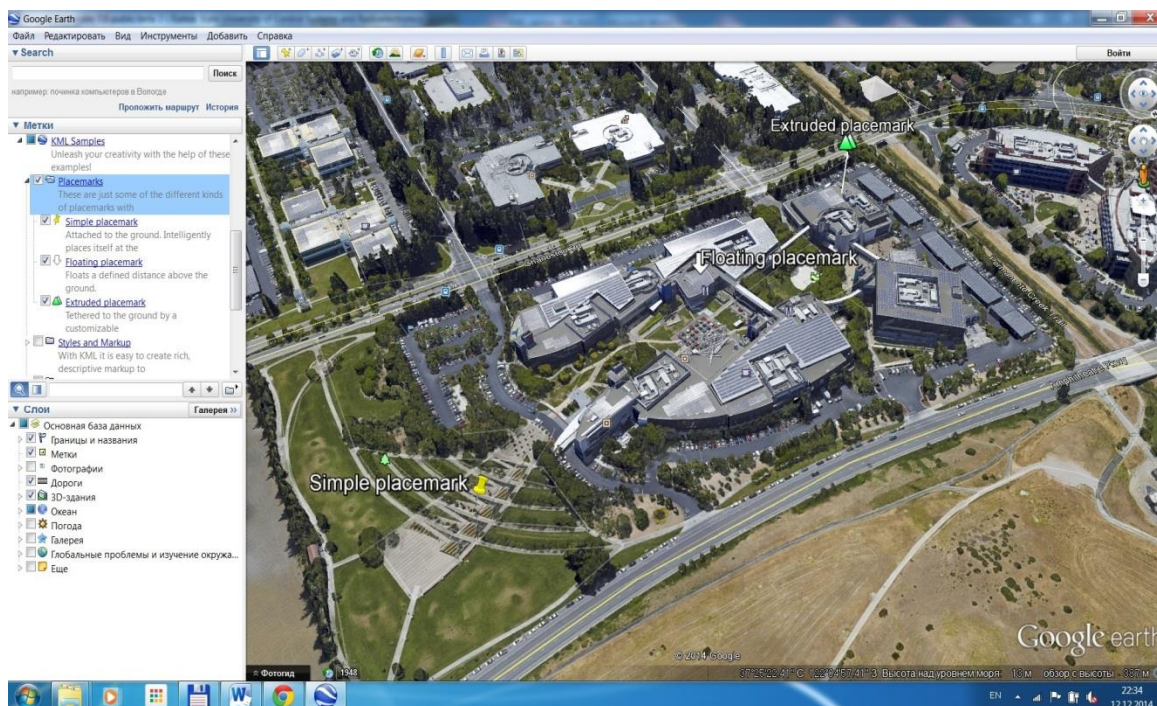


Рис. 1 Окно Google Earth после открытия файла KML\_Samples.kml

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 до-

бавление 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件трудно 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](http://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 <a href= ...> 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 &gt;, а символ < – как &lt;). 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><font color="red">Без ссылок на объекты текст <i>удобнее чи-
        тать</i> и <b>проще писать</b>.
        </font></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.

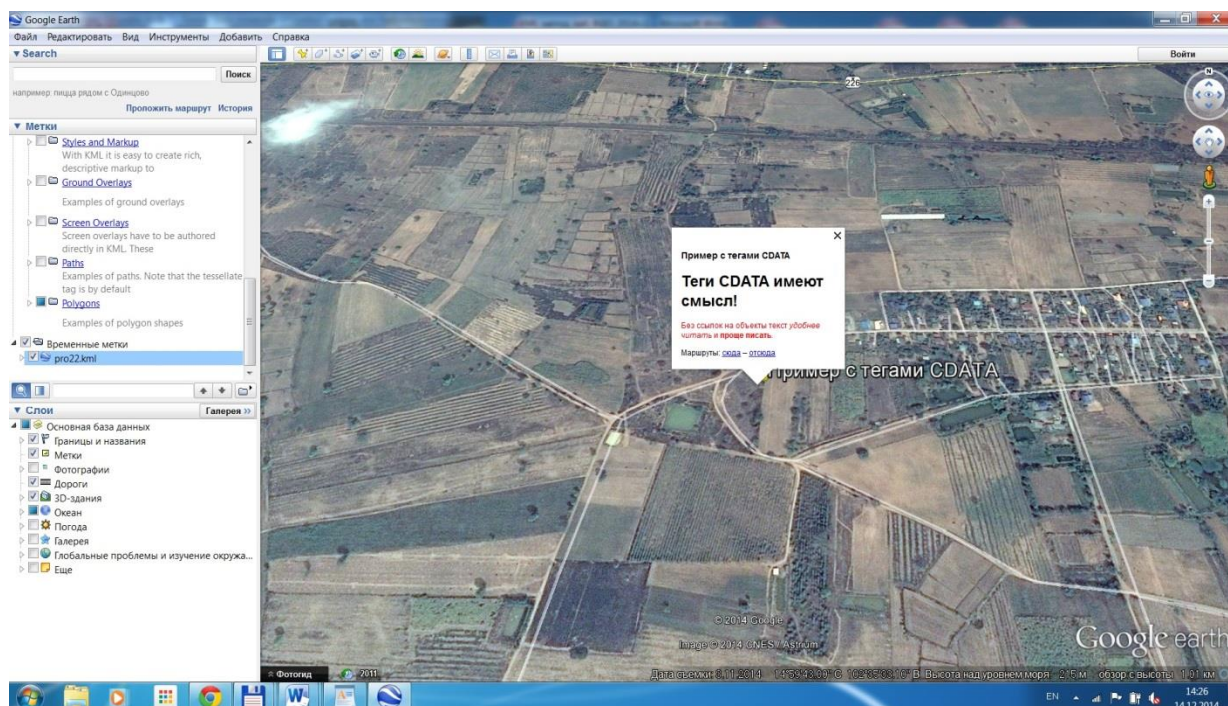


Рис. 2 Окно Google Earth после открытия файла с примером 1

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 (полигоны)*. 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>
```

Для векторизации границ полигонов (многоугольников) в программе Google Earth необходимо выбрать инструмент «Добавить многоугольник». Затем, в появившемся диалоге «Google Планета Земля – Создать: Многоугольник» (рис. 3) 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. 3).

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.

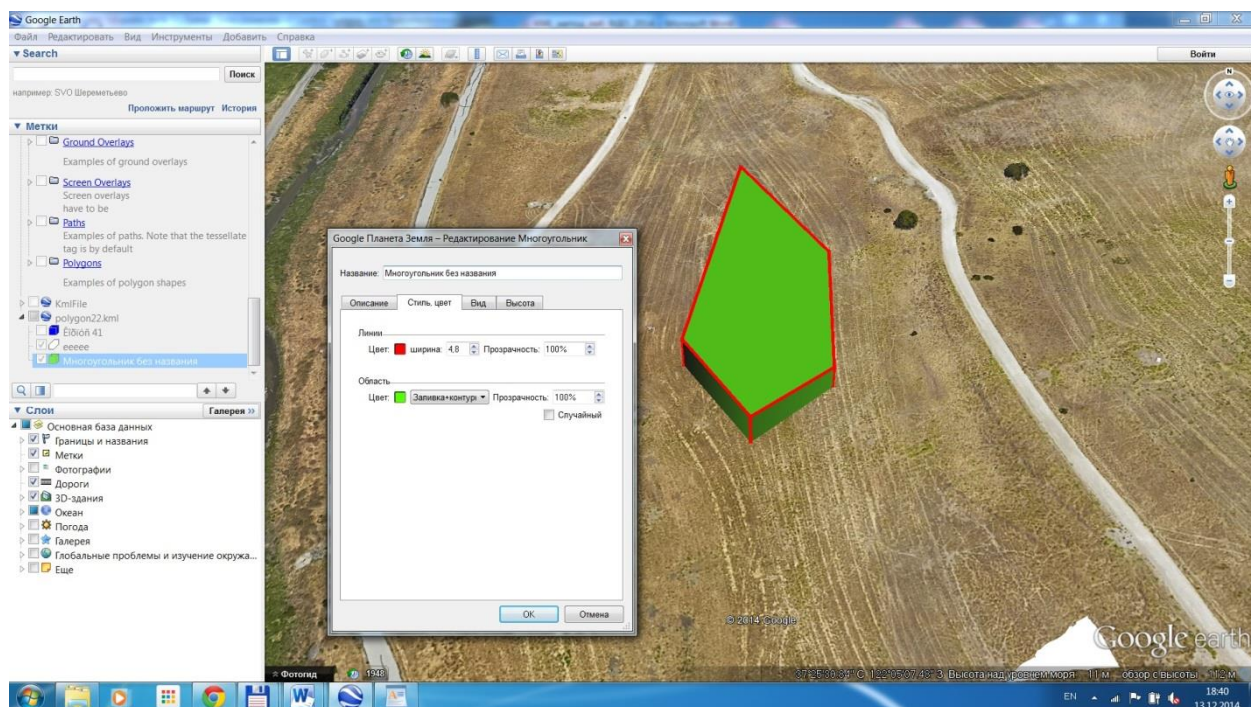


Рис. 3 Пример диалога с Google Earth в процессе создания полигона.

*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 ссылаться 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.

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.

Создайте элемент `<Style>` для значка метки в выделенном состоянии и присвойте ему идентификатор (в примере это `highlightPlacemark`).

Создайте элемент `<StyleMap>` и присвойте ему идентификатор (в примере это `exampleStyleMap`), на который будет ссылаться компонент `Placemark`.

Определите внутри элемента `<StyleMap>` ссылку на стиль `#normalPlacemark` для метки в обычном состоянии (`normal`).

Определите внутри элемента `<StyleMap>` ссылку на стиль `#highlightPlacemark` для выделенной метки (highlight).

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.j
pg
    </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.

### **3 ОФОРМЛЕНИЕ ОТЧЕТОВ О ВЫПОЛНЕНИИ ЛАБОРАТОРНЫХ РАБОТ**

При оформлении отчетов по текстовым работам (контрольным и лабораторным) следует руководствоваться требованиями образовательного стандарта вуза: ОС ТУСУР 01–2013. Работы студенческие по направлениям подготовки и специальностям технического профиля. Общие требования и правила оформления. – Режим доступа: [http://www.tusur.ru/export/sites/ru.tusur.new/ru/education/documents/inside/tech\\_01-2013\\_new.pdf](http://www.tusur.ru/export/sites/ru.tusur.new/ru/education/documents/inside/tech_01-2013_new.pdf)).

Отчет о выполнении лабораторной работы должен включать:

- 1) титульный лист (пример оформления приведен в приложении В);
- 2) содержание;
- 3) введение, в котором приводится формулировка задачи, определение цели и описание исходных данных;
- 4) основную часть, которая содержит:
  - 4.1) описание процесса управления, для которого планируется создать Хранилище Данных и функциональную модель для лабораторной работы «Разработка функциональной модели процесса создания хранилища данных»;
  - 4.2) Краткое описание размеченного географического объекта, текст kml-файла с его разметкой и сам kml-файла для лабораторной работы «Разметка географической информации на языке KML»;
- 5) заключение, которое содержит краткие выводы о проделанной работе;
- 6) приложения.

## ПРИЛОЖЕНИЕ А

### Варианты заданий к лабораторной работе

#### «Разработка функциональной модели процесса создания хранилища данных»

**Варианты предметных областей, предполагающих необходимость использования хранилищ данных в ходе процессов принятия решений при их управлении:**

1. Управление недвижимым имуществом города.
2. Управление недвижимым имуществом региона
3. Производство нефтехимической продукции.
4. Торговля автомобилями в регионе
5. Фармацевтическое производство на уровне региона.
6. Ведение аптечной сети.
7. Ведение региональной сети общественного питания.
8. Книготорговля на уровне республики.
9. Ведение сети продовольственных магазинов.
10. Ведение сети косметических магазинов.
11. Система здравоохранения региона.
12. Система муниципального здравоохранения.
13. Система среднего образования города.
14. Система среднего образования региона;
15. Система управлением среднего и малого предпринимательства региона.
16. Система управлением среднего и малого предпринимательства муниципального образования.
17. Управление общественным транспортом города.

18. Управление общественным транспортом региона.
19. Комитет оп управлению памятниками культуры региона.
20. Муниципальная музейная сеть.
21. Региональная музейная сеть.

## ПРИЛОЖЕНИЕ Б

### Варианты заданий к лабораторной работе

#### «Разметка географической информации на языке KML»

1. Маршрут от Вашего дома до корпуса ФЭТ ТУСУР. В случае иногороднего проживания точкам маршрута будут соответствовать населенные пункты, располагающиеся на маршруте. Для проживающих в Томске это будут точки на территории города.
2. Маршрут от главного корпуса ТУСУР до корпуса ФЭТ.
3. Маршрут Вашего летнего вояжа от дома до места отдыха.
4. Маршрут от Томска до Владивостока.
5. Маршрут от Вашего дома до места Вашего обычного места отдыха в выходные дни.
6. Маршрут для ознакомления потенциальных инвесторов с инвестиционно-привлекательными территориями Вашего города.
7. Маршрут от Вашего дома до места Вашей работы или обычного времяпровождения не в выходные дни (в случае надомной работы).
8. Маршрут от Вашего дома до места на земном шаре, где Вы хотите побывать.
9. Маршрут от Вашего дома до места на земном шаре, где Вы отдыхаете в зимнее время.
10. Маршрут от Вашего дома до места на земном шаре, где Вы отдыхаете в летнее время.
11. Маршрут для ознакомления потенциальных инвесторов с инвестиционно-привлекательными туристическими территориями Вашего города.

12. Маршрут для ознакомления впервые приехавших с достопримечательностями Вашего города.
13. Маршрут для ознакомления жителей старых районов с новыми достопримечательностями Вашего города.
14. Маршрут от Вашего дома до места на земном шаре, где бы Вы хотели жить.
15. Маршрут от Вашего дома до места на земном шаре, где бы Вы хотели отдыхать в летнее время.
16. Маршрут для ознакомления потенциальных инвесторов с инвестиционно-привлекательными территориями города Томска.
17. Маршрут для ознакомления потенциальных инвесторов с инвестиционно-привлекательными территориями Вашего региона.
18. Маршрут по памятным местам Вашего региона.
19. Маршрут от Вашего дома до средней школы, в которой Вы заканчивали обучение.
20. Маршрут к ближайшему от места Вашего проживания учреждению здравоохранения.

## ПРИЛОЖЕНИЕ В

### Форма титульного листа

Министерство образования и науки РФ

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Факультет дистанционного управления (ФДО)

**ТЕМА ЛАБОРАТОРНОЙ РАБОТЫ ПРОПИСНЫМИ БУКВАМИ**

Отчет о выполнении лабораторной работы  
по дисциплине «Информационные технологии в управлении»  
Вариант № *N*

Выполнил:  
Студент ФДО гр. *(номер)*  
*И. О. Фамилия*

*(дата)*

Проверил:  
доцент каф. АОИ ТУСУР,  
канд. техн. наук  
О. И. Жуковский

Год