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

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 Программная инженерия

О Т Ч Е Т

по учебной практике

Выполнил студент гр. РИС-19-1б

\_\_\_\_\_**\_\_\_\_**\_\_\_Миннахметов Э.Ю. \_**\_\_\_\_\_\_**

(фамилия, имя, отчество)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(подпись)

Проверил:

*\_\_\_старший преподаватель кафедры ИТАС Кузнецов Д.Б.\_\_*

*(должность, ФИО руководителя по практической подготовке от кафедры)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(оценка) (подпись)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

Пермь 2021

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ3

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ4

1.1 Ознакомление с инструментарием4

1.2 Ознакомление с псевдо-объектным представлением данных в системе META6

1.3 Анализ поставленной задачи9

2 ТЕХНОЛОГИЯ РЕАЛИЗАЦИИ11

2.1 Настройка индексации сервлета11

2.2 Оформление и дизайн веб-интерфейса11

2.3 Выполнение запросов к системе META13

ЗАКЛЮЧЕНИЕ17

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ18

ПРИЛОЖЕНИЕ А 19

ПРИЛОЖЕНИЕ Б 19

**В****ВЕДЕНИЕ**

**META**

**Цель:**

Разработка программного продукта для работы с таблицами баз данных (система учёта продаж автомобильного салона).

**Задачи:**

1. Проанализировать поставленную задачу (поиск информации, обработка поставленной задачи).
2. Разработать визуальную часть для удобства пользователя.
3. Разработать программную часть.

# **1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

# Суть данного раздела заключается в рассмотрении инструментария, который будет необходим при работе. Затем будут изложены принципы псевдо-объектного представления данных и их реализация в системе МЕТА. После чего будут описаны задания, поставленные компанией. Стоит начать с рассмотрения инструментария.

* 1. **Ознакомление с инструментарием**

В качестве операционной системы будет использоаться ***Linux*** – это корпаративное правило ООО «Комплексные системы» - выбор дистрибутива остается за программистом. В моем случае, будет использоваться ***KDE Neon*** из вкусовых соображений.

Программный код будет написан на языке ***Java*** и всем практикантам рекомендуется использовать среду разработки ***IntelliJ IDEA***, поскольку в ней был создан проект и иной выбор может повлечь проблемы совместимости. Выбор платной Ultimate или бесплатной Community версии также остается за программистом. В моем случае, это ***Ultimate*** по студенческой подписке.

Подключение к серверу будет производиться по ***SSH***, но прежде, чем подключиться к серверу, необходимо будет подключиться к ***VPN-сети*** – данная задача будет выполняться с помощью консольной программы ***OpenVPN***.

**Таким образом**, был представлен набор программного обеспечения, необходимый для комфортной работы. Далее будут рассмотрены принципы ***псевдо-объектного представления данных*** и их реализация в системе ***МЕТА***.

**1.2 Ознакомление с псевдо-объектным представлением данных в системе МЕТА**

Понятие "мета" используется для описания нескольких вещей:

а) наша основная библиотека доступа к данным - lib\_meta.jar;

б) родной вэб-интерфейс доступа к данным (без реализации бизнес-логики);

в) подход к представлению, хранению, доступу к данным.

Данные лежат в множестве реляционных баз, библиотека lib\_meta.jar (классы пакета appt.meta3.\*) обеспечивает доступ к ним в виде объектов с атрибутами. Каждый объект имеет уникальный идентификатор, сквозной для всех баз, номер типа, дату/время создания, смерти и автора объекта.

- Тип объекта - это число от 1 до 999, определяет набор атрибутов для всех объектов типа, параметры прав доступа к объектам, а так же расположение объектов типа в множестве реляционных баз.

- Уникальный идентификатор объекта (ID объекта) может быть 12- или 18- значным, id объекта однозначно определяет тип объекта (первые четыре знака минус 1000).

- Атрибуты объектов бывают 4 типов: строковый, численный, дата со временем, ссылка на другой объект. Для обращения к атрибутам объекта используется ID атрибута типа, он уникален в контексте типа, id атрибута типа имеет длину 10 знаков, id однозначно определяет номер типа, атрибутом объектов которого он является (первые четыре знака минус 1000), и тип атрибута (пятый знак). С атрибутами каждого типа лучше знакомиться на конкретных примерах объектов.

На этом объяснение того, что такое система МЕТА, будет прекращено, поскольку детали реализации являются коммерческой тайной компании, предоставляющей место для практики. Далее будут рассмотрены задачи, которые поставила компания перед командой практикантов.

**1.3 Анализ поставленной задачи**

Для выполнения задачи потребуется скопировать проект с системы управления версиями Subversion (SVN) с сервера компании, к сети которого будет произведено подключение по VPN. Детали данных действий несут тайный характер, поэтому следует перейти к непосредственному рассмотрению некоторых из задач:

1. Выбрать все города (тип 5 1000004) России 100410000050, где 1000101 Городвылета = Да. На экран вывести таблице, в первой колонке по алфавиту идут территории, а во втором столбце по алфавиту города в этих территориях: Название области 1005101368, Название города 1000098. (Сделать возможность менять страну в форме).
2. Выбрать всех Туристов (тип 23), которые родились позже 01.01.1990. На экран для каждого туриста вывести в формате JSON: ФИО туриста, дату рождения и экскурсии массивом. 1000152 дата рождения 1023422081 экскурсии массивом 1000144 1000146 1000147 ФИО (Сделать возможность менять дату рождения в url)
3. Сделать форму для создания описания экскурсии 506. Название 1506410000 Описание краткое 1506410282 Регионы 1506923461 (на город) Дополнительно оплачиваются входные билеты 1506223120 Бронировать у партнера 1506910189 (ссылка на партнера 158 тип, название 1001211) Тип (экскурсия - 0, билет - 1, спорт - 2, прокат - 3, услуга - 4, СПА - 5, авиация - 6) 1506310181
4. Сделать возможность удалять объекты 506 типа в форме - заносим id объекта, который надо удалить, по кнопке Удалить - объект удаляется

Такова трактовка задания, детали реализации которого будут рассмотрены в следующем разделе.

# **2 ТЕХНОЛОГИЯ РЕАЛИЗАЦИИ**

В данном разделе будет показан код сервлета на языке Java, который, в зависимости от GET-параметров, будет выполнять запросы к системе МЕТА, затем заворачивать их в требуемый формат – в основном, это HTML, но одно из заданий будет выводить результат в формате JSON.

**2.1 Настройка индексации сервлета**

Листинг 2.1 – ваываываыв

<servlet>

<servlet-name>EldarServlet</servlet-name>

<servlet-class>intern.EldarServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>EldarServlet</servlet-name>

<url-pattern>/eldar</url-pattern>

</servlet-mapping>

ваыаываы

**Таким образом**, был проиндексирован мой сервлет и при обращении к серверу с запросом веб-страницы “/eldar”, будет выводиться результат работы написанного мной сервлета EldarServlet.

Поскольку 3 из 4 заданий будут выводить результат в HTML, мною было решено заранее стилизовать и принятые решения будут описаны далее.

**2.2 Оформление и дизайн веб-интерфейса**

Листинг 2.1 – ваываываы

public void printHtml(String title, String text) {  
 response.setContentType("text/html; charset=UTF-8");  
 out.printf("<!doctype html><html><head><title>%s</title><style>" +  
 "table { margin: 0px auto; }" +  
 "table.form tr td:first-child { text-align:right; }" +  
 "table.form tr td:last-child { text-align:left; }" +  
 "table.data { border: solid 1px #ccc; border-spacing: 3px;" +  
 "border-collapse: collapse; margin-bottom:10px; }" +  
 "table.data tr:first-child { font-weight:bold; }" +  
 "table.data td { border: solid 1px #ccc; padding: 5px; }" +  
 "form { margin: 0px auto; margin-bottom:20px; text-align:center;" +  
 "padding:10px; }" +  
 "</style></head>" +  
 "<body><div style=\"text-align:center;margin-bottom:20px;\">" +  
 "<a href=\"?task=1\">Один</a> | " +  
 "<a href=\"?task=2\">Два</a> | " +  
 "<a href=\"?task=3\">Три</a> | " +  
 "<a href=\"?task=4\">Четыре</a> | " +  
 "<a href=\"?task=5\">Пять</a> | " +  
 "<a href=\"?task=6\">Шесть</a> | " +  
 "<a href=\"?task=7\">Семь</a> | " +  
 "<a href=\"?task=8\">Восемь</a>" +  
 "</div>%s</body></html>", title, text);  
 }

аываываы

**2.3 Выполнение запросов к системе МЕТА**

чсммв

**ЗАКЛЮЧЕНИЕ**

В процессе выполнения практического проекта были использованы среда разработки Sap, язык программирования ABAP.

Были выполнены следующие задачи: проведён анализ предметной области, разработка визуализации программы, реализация программы для работы с таблицами баз данных.

Цель данной практической работы, а именно, разработка программного продукта для работы с таблицами баз данных (система учёта продаж автомобильного салона) была достигнута.

**СПИСОК ИСТОЧНИКОВ**

1. Кале Вивек. Внедрение SAP R/3: Руководство для менеджеров и инженеров. 2006 – 505 c.

2. BC400 – Введение в ABAP – инструментальные средства. SAP NetWeaver. 2005 – 361 c.

3. BC401 – ABAP Объекты. SAP NetWeaver. 2009 – 448 c.

4.Хомоненко А. Д. Базы данных: Учебник для высших учебных заведений/  В.М. Цыганков, М. Г. Мальцев, под ред. проф. А. Д. Хомоненко— СПб.:  КОРОНА принт, 2002 — 672 с.

5.Дейт К. Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом “Вильямс”, 2005. — 1328 с.

**ПРИЛОЖЕНИЕ А**

<?xml version="1.0" encoding="ISO-8859-1"?>

<web-app>

<servlet>

<servlet-name>InokovaServlet</servlet-name>

<servlet-class>intern.InokovaServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>InokovaServlet</servlet-name>

<url-pattern>/inokovaServlet</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>EldarServlet</servlet-name>

<servlet-class>intern.EldarServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>EldarServlet</servlet-name>

<url-pattern>/eldar</url-pattern>

</servlet-mapping>

</web-app>

**ПРИЛОЖЕНИЕ Б**

package intern;  
  
import appt.meta3.Ob0;  
import appt.meta3.Obb;  
import org.apache.commons.lang3.math.NumberUtils;  
  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.util.\*;  
import java.util.function.BiConsumer;  
  
public class EldarServlet extends HttpServlet {  
 private PrintWriter out;  
 private ResourceBundle mains;  
 private HttpServletRequest request;  
 private HttpServletResponse response;  
 private final int myId = 1000360;  
  
 private static final Map<String, String> *typeMap* = new HashMap<>();  
 private static final Map<String, String> *yesnoMap* = new HashMap<>();  
  
 static {  
 *typeMap*.put("", "-");  
 *typeMap*.put("0", "Экскурсия");  
 *typeMap*.put("1", "Билет");  
 *typeMap*.put("2", "Спорт");  
 *typeMap*.put("3", "Прокат");  
 *typeMap*.put("4", "Услуга");  
 *typeMap*.put("5", "СПА");  
 *typeMap*.put("6", "Авиация");  
 *typeMap*.put("8", "Концерт");  
 *yesnoMap*.put("", "-");  
 *yesnoMap*.put("1", "Да");  
 *yesnoMap*.put("0", "Нет");  
 }  
  
 public void doPost(HttpServletRequest req, HttpServletResponse res) throws IOException {  
 initialize(req, res);  
 String task = request.getParameter("task");  
 try {  
 if ("7".equals(task)) {  
 task7post();  
 }  
 } catch (Exception e) {  
 printHtml("Exception!", String.*format*("<h1>%s: %s</h1>",  
 e.getClass().getName(), e.getMessage()));  
 }  
 out.flush();  
 }  
  
 public void doGet(HttpServletRequest req, HttpServletResponse res) throws IOException {  
 initialize(req, res);  
 try {  
 int task = Integer.*parseInt*(request.getParameter("task"));  
 if(task < 1 || task > 8) {  
 throw new NumberFormatException();  
 }  
 if(task == 1) {  
 task1();  
 } else if(task == 2) {  
 task2();  
 } else if(task == 3) {  
 task3();  
 } else if(task == 4) {  
 task4();  
 } else if(task == 5) {  
 task5();  
 } else if(task == 6) {  
 task6();  
 } else if(task == 7) {  
 task7();  
 } else {  
 task8();  
 }  
 } catch (NumberFormatException e) {  
 printHtml("Hello", "<h1>Привет Sirius!</h1>");  
 } catch (Exception e) {  
 printHtml("Exception!", String.*format*("<h1>%s: %s</h1>",  
 e.getClass().getName(), e.getMessage()));  
 }  
 out.flush();  
 }  
  
 public void initialize(HttpServletRequest req, HttpServletResponse res) throws IOException {  
 request = req;  
 response = res;  
 response.setCharacterEncoding("UTF-8");  
 request.setCharacterEncoding("UTF-8");  
 out = res.getWriter();  
 mains = ResourceBundle.*getBundle*("intern\_main", new Locale("ru", "RU"));  
 }  
  
 public void task1() throws Exception {  
 Map<String, String> regions = getRegions("100410000050");  
 Obb filter = Ob0.*createFilter*(5);  
 String regionId = request.getParameter("region");  
 if(regionId == null || regionId.equals("")) {  
 regionId = "100518301512";  
 }  
 Ob0.*addCondition*(filter, 1005101368, Ob0.ComparisonType.*EQ*, regionId);  
 Obb[] cities = Ob0.*getSrcObs*(mains, filter, 0, 0);  
 StringBuilder text = new StringBuilder("<form action=\"#\" method=\"get\"><table class=\"form\"><tr><td>" +  
 "Регион</td><td><select name=\"region\">");  
 for(String key : regions.keySet()) {  
 text.append(String.*format*("<option%s value=\"%s\">%s</option>",  
 key.equals(regionId) ? " selected" : "", key, regions.get(key)));  
 }  
 text.append("</select></td></tr></table><input type=\"submit\" value=\"Выбрать\">" +  
 "<input type=\"hidden\" name=\"task\" value=\"1\"></form>");  
 int i = 0;  
 text.append("<table class=\"data\"><tr><td>№</td><td>Город</td></tr>");  
 for(Obb city : cities) {  
 text.append(String.*format*("<tr><td>%d</td><td>%s</td></tr>",  
 ++i, Ob0.*getAt*(city, "1000098")));  
 }  
 text.append("</table>");  
 printHtml("Города России", text.toString());  
 }  
  
 public Map<String, String> getRegions(String countryId) throws Exception {  
 Obb filter = Ob0.*createFilter*(5);  
 Ob0.*addCondition*(filter, 1000004, Ob0.ComparisonType.*EQ*, countryId);  
 Ob0.*addCondition*(filter, 1005101368, Ob0.ComparisonType.*NEQ*, "");  
 Obb[] cities = Ob0.*getSrcObs*(mains, filter, 0, 0);  
 Map<String, String> regions = new TreeMap<>();  
 for(Obb city : cities) {  
 String regionId = Ob0.*getAt*(city, 1005101368);  
 if(!regions.containsKey(regionId)) {  
 regions.put(regionId, Ob0.*getZn*(mains, regionId, 1000098, 4));  
 }  
 }  
 return regions;  
 }  
  
 public void task2() throws Exception {  
 String countryId = request.getParameter("country");  
 if(countryId == null || countryId.equals("")) {  
 countryId = "100410000050";  
 }  
 Obb filter = Ob0.*createFilter*(5);  
 Ob0.*addCondition*(filter, 1000004, Ob0.ComparisonType.*EQ*, countryId);  
 Ob0.*addCondition*(filter, 1000101, Ob0.ComparisonType.*EQ*, "Да");  
 Obb[] cities = Ob0.*getSrcObs*(mains, filter, 0, 0);  
 Obb[] countries = Ob0.*getSrcObs*(mains, Ob0.*createFilter*(4), 0, 0);  
 Map<String, String> regions = getRegions(countryId);  
 Map<String, Set<String>> distribution = regionDistribute(cities);  
 StringBuilder text = new StringBuilder();  
  
 text.append("<form action=\"#\" method=\"get\"><table class=\"form\"><tr><td>" +  
 "Страна</td><td><select name=\"country\">");  
 for(Obb country : countries) {  
 text.append(String.*format*("<option%s value=\"%s\">%s</option>",  
 country.id.equals(countryId) ? " selected" : "", country.id, country.getAt("1000000")));  
 }  
 text.append("</select></td></tr></table><input type=\"submit\" value=\"Выбрать\">" +  
 "<input type=\"hidden\" name=\"task\" value=\"2\"></form>");  
  
 text.append("<table class=\"data\"><tr><td>№</td><td>Область</td><td>Города</td></tr>");  
 int i = 0;  
 for(String key : distribution.keySet()) {  
 if(regions.containsKey(key)) {  
 text.append(String.*format*("<tr><td>%d</td><td>%s</td><td>%s</td></tr>",  
 ++i, regions.get(key), String.*join*(", ", distribution.get(key))));  
 }  
 }  
 text.append("</table>");  
  
 printHtml("Города России", text.toString());  
 }  
  
 public Map<String, Set<String>> regionDistribute(Obb[] cities) {  
 Map<String, Set<String>> distribution = new TreeMap<>();  
 for(Obb city : cities) {  
 String regionId = Ob0.*getAt*(city, 1005101368);  
 if(!distribution.containsKey(regionId)) {  
 distribution.put(regionId, new TreeSet<>());  
 }  
 distribution.get(regionId).add(Ob0.*getAt*(city, 1000098));  
 }  
 return distribution;  
 }  
  
 public void task3() throws Exception {  
 response.setContentType("application/json; charset=UTF-8");  
 String birthday = request.getParameter("bd");  
 if(birthday == null || birthday.equals("")) {  
 birthday = "01.01.1990";  
 }  
 Obb filter = Ob0.*createFilter*(23);  
 Ob0.*addCondition*(filter, 1000152, Ob0.ComparisonType.*GT*, birthday);  
 Obb[] tourists = Ob0.*getSrcObs*(mains, filter, 0, 0);  
 out.print("[");  
 int i = 0;  
 for(Obb tourist : tourists) {  
 out.printf("{\"id\":\"%s\",\"nm\":\"%s %s %s\",\"bd\":\"%s\",\"trs\":[",  
 tourist == null ? "null" : tourist.id,  
 Ob0.*getAt*(tourist, 1000144),  
 Ob0.*getAt*(tourist, 1000146),  
 Ob0.*getAt*(tourist, 1000147),  
 Ob0.*getAt*(tourist, 1000152));  
 String[] tours = Ob0.*getAt*(tourist, 1023422081).split("#");  
 int j = 0;  
 for(String tour : tours) {  
 out.printf("\"%s\"%s", tour, ++j < tours.length ? "," : "");  
 }  
 out.printf("]}%s", ++i < tourists.length ? "," : "");  
 }  
 out.print("]");  
 }  
  
 public void task4() throws Exception {  
 String agentName = request.getParameter("agent");  
 String categoryName = request.getParameter("category");  
 if(agentName == null || agentName.equals("")) {  
 agentName = "КонтрАг";  
 }  
 if(categoryName == null || categoryName.equals("")) {  
 categoryName = "Концерты";  
 }  
 Obb filter = Ob0.*createFilter*(36);  
 Ob0.*addCondition*(filter, new int[]{1036922797, 1317100000}, Ob0.ComparisonType.*EQ*, agentName);  
 Ob0.*addCondition*(filter, new int[]{1036900082, 1162100000}, Ob0.ComparisonType.*EQ*, categoryName);  
 Obb[] costs = Ob0.*getSrcObs*(mains, filter, 0, 0);  
 Obb[] agents = Ob0.*getSrcObs*(mains, Ob0.*createFilter*(317), 0, 0);  
 Obb[] categories = Ob0.*getSrcObs*(mains, Ob0.*createFilter*(162), 0, 0);  
 Arrays.*sort*(costs, (Obb left, Obb right) -> right.data\_n.compareTo(left.data\_n));  
 costs = Arrays.*copyOfRange*(costs, Math.*max*(0, costs.length - 50), costs.length);  
 StringBuilder text = new StringBuilder();  
 text.append("<form action=\"#\" method=\"get\"><table class=\"form\"><tr><td>" +  
 "Контрагент</td><td><select name=\"agent\">");  
 for(Obb agent : agents) {  
 String an = agent.getAt("1317100000");  
 text.append(String.*format*("<option%s>%s</option>",  
 an.equals(agentName) ? " selected" : "", an));  
 }  
 text.append("</select></td></tr><tr><td>Категория</td><td><select name=\"category\">");  
 for(Obb category : categories) {  
 String cn = category.getAt("1162100000");  
 text.append(String.*format*("<option%s>%s</option>",  
 cn.equals(categoryName) ? " selected" : "", cn));  
 }  
 text.append("</select></td><tr></table><input type=\"submit\" value=\"Выбрать\">" +  
 "<input type=\"hidden\" name=\"task\" value=\"4\"></form>" +  
 "<table class=\"data\"><tr><td>№</td><td>Название</td><td>Категория</td><td>Адрес</td></tr>");  
 int i = 0;  
 for(Obb cost : costs) {  
 text.append(String.*format*("<tr><td>%d</td><td>%s</td><td>%s</td><td>%s</td></tr>",  
 ++i, cost.getAt("1036423021"),  
 *yesnoMap*.get(cost.getAt("1036200042")),  
 cost.getAt("1036410028")));  
 }  
 printHtml("Экскурсии", text.append("</table>").toString());  
 }  
  
 public void task5() throws Exception {  
 Obb[] costs = Ob0.*getSrcObs*(mains, Ob0.*createFilter*(36), 0, 0);  
 List<Obb> list = Arrays.*asList*(costs);  
 List<Obb> array = new ArrayList<>(list);  
 List<Obb> linked = new LinkedList<>(list);  
 StringBuilder text = new StringBuilder();  
 BiConsumer<List<Obb>, String> consumer = (List<Obb> lst, String listType) -> {  
 int i = 0;  
 text.append("<div>");  
 long time = System.*currentTimeMillis*();  
 for(Obb item : lst) {  
 text.append(String.*format*("[%d : %s]%s", ++i, item.id, i == lst.size() ? "" : ", "));  
 }  
 time = System.*currentTimeMillis*() - time;  
 text.append(String.*format*("<br>%s - %d ms</div>", listType, time));  
 };  
 consumer.accept(array, "ArrayList");  
 consumer.accept(linked, "LinkedList");  
 printHtml("Временное сравнение", text.toString());  
 }  
  
 public void task6() throws Exception {  
 String type = request.getParameter("type");  
 String code = request.getParameter("code");  
 String address = request.getParameter("address");  
 String category = request.getParameter("category");  
 String agentId = request.getParameter("agent");  
  
 type = NumberUtils.*isNumber*(type) && Integer.*parseInt*(type) >= 0  
 && Integer.*parseInt*(type) <= 8 && Integer.*parseInt*(type) != 7 ? type : "";  
 code = code == null ? "" : code;  
 address = address == null ? "" : address;  
 category = category == null || !(category.equals("1") || category.equals("0")) ? "" : category;  
 agentId = agentId == null ? "" : agentId;  
  
 Obb filter = Ob0.*createFilter*(36);  
 if(!type.equals("")) {  
 Ob0.*addCondition*(filter, 1036200042, Ob0.ComparisonType.*EQ*, type);  
 }  
 if(!code.equals("")) {  
 Ob0.*addCondition*(filter, 1036423021, Ob0.ComparisonType.*EQ*, code);  
 }  
 if(!address.equals("")) {  
 Ob0.*addCondition*(filter, 1036410028, Ob0.ComparisonType.*EQ*, address);  
 }  
 if(!category.equals("")) {  
 Ob0.*addCondition*(filter, 1162200125, Ob0.ComparisonType.*EQ*, category);  
 }  
 if(!agentId.equals("")) {  
 Ob0.*addCondition*(filter, 1036922797, Ob0.ComparisonType.*EQ*, agentId);  
 }  
 Obb[] costs = Ob0.*getSrcObs*(mains, filter, 0, 0);  
 Obb[] agents = Ob0.*getSrcObs*(mains, Ob0.*createFilter*(317), 0, 0);  
  
 Map<String, String> agentMap = new HashMap<>();  
 agentMap.put("", "-");  
 for(Obb agent : agents) {  
 agentMap.put(agent.id, agent.getAt("1317100000"));  
 }  
  
 StringBuilder text = new StringBuilder();  
 text.append("<form action=\"#\" method=\"get\"><table class=\"form\"><tr><td>" +  
 "Тип</td><td><select name=\"type\">");  
 for(String key : *typeMap*.keySet()) {  
 text.append(String.*format*("<option%s value=\"%s\">%s</option>",  
 type.equals(key) ? " selected" : "", key, *typeMap*.get(key)));  
 }  
 text.append("</select></td></tr><tr><td>Код</td><td><input name=\"code\" type=\"text\" value=\"\"></td></tr>" +  
 "<tr><td>Адрес</td><td><input name=\"address\" type=\"text\" value=\"\"></td></tr>" +  
 "<tr><td>Категория</td><td><select name=\"category\">");  
 for(String key : *yesnoMap*.keySet()) {  
 text.append(String.*format*("<option%s value=\"%s\">%s</option>",  
 category.equals(key) ? " selected" : "", key, *yesnoMap*.get(key)));  
 }  
 text.append("</select></td></tr><tr><td>Контрагент</td><td><select name=\"agent\">");  
 for(String key : agentMap.keySet()) {  
 text.append(String.*format*("<option%s value=\"%s\">%s</option>",  
 agentId.equals(key) ? " selected" : "", key, agentMap.get(key)));  
 }  
 text.append("</select></td></tr></table><input type=\"submit\" value=\"Выбрать\">" +  
 "<input type=\"hidden\" name=\"task\" value=\"6\"></form>" +  
 "<table class=\"data\"><tr><td>№</td><td>Тип</td><td>Название</td><td>Адрес</td><td>Да/Нет</td>" +  
 "<td>Контрагент</td></tr>");  
 int i = 0;  
 for(Obb cost : costs) {  
 text.append(String.*format*("<tr><td>%d</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td></tr>",  
 ++i, *typeMap*.get(cost.getAt("1036200042")),  
 cost.getAt("1036423021"),  
 cost.getAt("1036410028"),  
 *yesnoMap*.get(cost.getAt("1162200125")),  
 agentMap.get(cost.getAt("1036922797"))));  
 }  
 printHtml("Экскурсии", text.append("</table>").toString());  
 }  
  
 public void task7() throws Exception {  
 Map<String, String> regions = new HashMap<>();  
 Map<String, String> partners = new HashMap<>();  
 regionsAndPartners(regions, partners);  
  
 StringBuilder text = new StringBuilder();  
 text.append("<form method=\"post\" action=\"#\"><table class=\"form\">" +  
 "<tr><td>Название</td><td><input type=\"text\" name=\"name\"></td></tr>" +  
 "<tr><td>Описание</td><td><textarea name=\"desc\"></textarea></td></tr>" +  
 "<tr><td>Регион</td><td><select name=\"region\">");  
 for(String regionId : regions.keySet()) {  
 text.append(String.*format*("<option value=\"%s\">%s</option>", regionId, regions.get(regionId)));  
 }  
 text.append("</select></td></tr>" +  
 "<tr><td>Доп. оплата</td><td><input type=\"text\" name=\"cost\"></td></tr>" +  
 "<tr><td>Бронирование у партнера</td><td><select name=\"partner\">");  
 for(String partnerId : partners.keySet()) {  
 text.append(String.*format*("<option value=\"%s\">%s</option>", partnerId, partners.get(partnerId)));  
 }  
 text.append("</select></td></tr>" +  
 "<tr><td>Тип</td><td><select name=\"type\">");  
 for(String typeId : *typeMap*.keySet()) {  
 text.append(String.*format*("<option value=\"%s\">%s</option>", typeId, *typeMap*.get(typeId)));  
 }  
 text.append("</select></td></tr>" +  
 "</table><input type=\"hidden\" name=\"task\" value=\"7\">" +  
 "<input type=\"submit\" value=\"Создать\"></form>");  
 printDescs(text, regions, partners);  
 }  
  
 public void task7post() throws Exception {  
 String name = request.getParameter("name");  
 String desc = request.getParameter("desc");  
 String region = request.getParameter("region");  
 String cost = request.getParameter("cost");  
 String partner = request.getParameter("partner");  
 String type = request.getParameter("type");  
  
 name = name == null ? "" : name;  
 desc = desc == null ? "" : desc;  
 region = region == null ? "-" : region;  
 cost = !NumberUtils.*isNumber*(cost) ? "" : cost;  
 partner = partner == null ? "-" : partner;  
 type = !*typeMap*.containsKey(type) ? "-" : type;  
  
 if(!name.equals("") && !desc.equals("") && !region.equals("-") &&  
 !cost.equals("") && !partner.equals("-") && !type.equals("-")) {  
 Obb ob = new Obb(506);  
 ob.id\_user = myId;  
 Ob0.*addAt*(ob, "1506410000", name);  
 Ob0.*addAt*(ob, "1506410282", desc);  
 Ob0.*addAt*(ob, "1506923461", region);  
 Ob0.*addAt*(ob, "1506223120", cost);  
 Ob0.*addAt*(ob, "1506910189", partner);  
 Ob0.*addAt*(ob, "1506310181", type);  
 Ob0.*addOb*(mains, ob);  
 }  
 response.sendRedirect("/eldar?task=7");  
 }  
  
 public void task8() throws Exception {  
 String descId = request.getParameter("did");  
 descId = descId == null ? "" : descId;  
 if(!descId.equals("")) {  
 Ob0.*delOb*(mains, descId, myId);  
 }  
 Map<String, String> regions = new HashMap<>();  
 Map<String, String> partners = new HashMap<>();  
 regionsAndPartners(regions, partners);  
 StringBuilder text = new StringBuilder();  
 text.append("<form method=\"get\" action=\"#\">ID <input type=\"text\" name=\"did\">" +  
 "<input type=\"submit\" value=\"Удалить\"><input type=\"hidden\" name=\"task\" value=\"8\"></form>");  
 printDescs(text, regions, partners);  
 }  
  
 public void regionsAndPartners(Map<String, String> regions,  
 Map<String, String> partners) throws Exception {  
 Obb filter = Ob0.*createFilter*(5);  
 Ob0.*addCondition*(filter, 1000004, Ob0.ComparisonType.*EQ*, "100410000050");  
 Ob0.*addCondition*(filter, 1005101368, Ob0.ComparisonType.*NEQ*, "");  
 Obb[] cities = Ob0.*getSrcObs*(mains, filter, 0, 0);  
 Obb[] prtns = Ob0.*getSrcObs*(mains, Ob0.*createFilter*(158), 0, 0);  
 regions.put("", "-");  
 partners.put("", "-");  
 for(Obb city : cities) {  
 regions.put(city.id, city.getAt("1000098"));  
 }  
 for(Obb partner : prtns) {  
 partners.put(partner.id, partner.getAt("1001211"));  
 }  
 }  
  
 public void printDescs(StringBuilder text, Map<String, String> regions,  
 Map<String, String> partners) throws Exception {  
 Obb[] descs = Ob0.*getSrcObs*(mains, Ob0.*createFilter*(506), 0, 0);  
 text.append("<table class=\"data\"><tr><td>№</td><td>ID</td><td>Название</td><td>Описание</td>" +  
 "<td>Регион</td><td>Доп. оплаты</td><td>Бронирование у парнера</td><td>Тип</td></tr>");  
 int i = 0;  
 for(Obb desc : descs) {  
 text.append(String.*format*("<tr><td>%d</td><td>%s</td><td>%s</td><td>%s</td>" +  
 "<td>%s</td><td>%s</td><td>%s</td><td>%s</td></tr>",  
 ++i, (desc.id\_user == myId ? "<b style=\"color:green;\">" + desc.id + "<b>" : desc.id),  
 desc.getAt("1506410000"),  
 desc.getAt("1506410282"),  
 regions.get(desc.getAt("1506923461")),  
 desc.getAt("1506223120"),  
 partners.get(desc.getAt("1506910189")),  
 *typeMap*.get(desc.getAt("1506310181"))));  
 }  
 text.append("</table>");  
 printHtml("Описания экскурсий", text.toString());  
 }  
  
 public void printHtml(String title, String text) {  
 response.setContentType("text/html; charset=UTF-8");  
 out.printf("<!doctype html><html><head><title>%s</title><style>" +  
 "table { margin: 0px auto; }" +  
 "table.form tr td:first-child { text-align:right; }" +  
 "table.form tr td:last-child { text-align:left; }" +  
 "table.data { border: solid 1px #ccc; border-spacing: 3px;" +  
 "border-collapse: collapse; margin-bottom:10px; }" +  
 "table.data tr:first-child { font-weight:bold; }" +  
 "table.data td { border: solid 1px #ccc; padding: 5px; }" +  
 "form { margin: 0px auto; margin-bottom:20px; text-align:center;" +  
 "padding:10px; }" +  
 "</style></head>" +  
 "<body><div style=\"text-align:center;margin-bottom:20px;\">" +  
 "<a href=\"?task=1\">Один</a> | " +  
 "<a href=\"?task=2\">Два</a> | " +  
 "<a href=\"?task=3\">Три</a> | " +  
 "<a href=\"?task=4\">Четыре</a> | " +  
 "<a href=\"?task=5\">Пять</a> | " +  
 "<a href=\"?task=6\">Шесть</a> | " +  
 "<a href=\"?task=7\">Семь</a> | " +  
 "<a href=\"?task=8\">Восемь</a>" +  
 "</div>%s</body></html>", title, text);  
 }  
}