



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-19-16

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

Введение	3
1 Анализ предметной области.....	4
1.1 Ознакомление с компанией ООО «Комплексные системы»	4
1.2 Псевдо-объектное представление данных в системе МЕТА	5
1.3 Анализ поставленной задачи.....	6
2 Технология реализации.....	8
2.1 Настройка индексации сервлета	8
2.2 Оформление кода и дизайн веб-интерфейса	8
2.3 Выполнение запросов к системе МЕТА	13
Заключение.....	18
Список использованных источников	19
Приложение А.....	20
Приложение Б	21

Введение

ООО «Комплексные системы» — это группа компаний. Во всех этих компаниях используется собственный продукт – система META. Она позволяет работать с такими базами данных, как PostgreSQL и NoSQL-СУБД Redis. В системе META используется собственная нотация представления данных, называемая Псевдо-объектным представлением данных.

Цель практической работы: разработка программного продукта для работы с системой META в СУБД PostgreSQL и NoSQL-СУБД Redis.

Для исполнения поставленной цели, ее необходимо разбить на следующие **задачи**:

- 1) предоставить ознакомительные данные о предприятии;
- 2) изучить теорию по Псевдо-объектному представлению данных в системе META, а также его реализацию в Java-библиотеке lib_meta.jar;
- 3) рассмотреть задачи, предоставленные компанией;
- 4) написать сервлет для обработки запросов браузера;
- 5) разработать дизайн для вывода сервлета;
- 6) решить поставленные задачи.

1 Анализ предметной области

1.1 Ознакомление с компанией ООО «Комплексные системы»

ООО «Комплексные системы» - крупная, стабильная, развивающаяся компания, которая занимается разработкой информационных систем и программных продуктов более 20 лет. Компания обеспечиваем комплексный подход к автоматизации больших систем высокого класса и уровня сложности за счёт встроенных механизмов саморегуляции и уникальной модели организации и хранения данных. ^[1]

Команда «Комплексных систем» — это уникальные специалисты, которые создают программное обеспечение и мобильные приложения различного уровня сложности и направленности. Научные кадры компании представлены кандидатами физико-математических и технических наук, докторами технических наук, профессорами. Команда «Комплексных систем» - профессионалы с большим опытом работы в разных сферах бизнеса, и она может с легкостью реализовать ваши самые смелые и амбициозные идеи. ^[1]

Компанией разработана система МЕТА, которая показала свою эффективность во многих проектах. ^[2]

МЕТА — это совокупность методов и технических приемов, обеспечивающих глубокий комплексный подход к автоматизации систем высокого класса и уровня сложности за счет встроенных механизмов саморегуляции. ^[2]

Система МЕТА обладает механизмами балансировки нагрузки, резервирования данных и анализа собственной инфраструктуры, что обеспечивает стабильную непрерывную работу системы и минимизирует затраты на ее поддержку и обслуживание. ^[2]

В компании используется следующий стек технологий для разработки:

- Язык программирования Java;
- Система управления базами данных (СУБД) PostgreSQL;
- NoSQL-СУБД Redis;
- Библиотека для работы с базами данных в системе МЕТА lib_meta.jar;

- Интегрированная среда разработки IntelliJ IDEA;
- Контейнер сервлетов Apache Tomcat;
- Сервер Nginx;
- Система контроля версий Subversion (SVN).

Далее будет рассмотрена система META – элемент стека разработки, который входит во внутреннее использование ООО «Комплексные системы».

1.2 Псевдо-объектное представление данных в системе META

Понятие "мета" используется для описания нескольких вещей:

- 1) основная библиотека доступа к данным - `lib_meta.jar`;
- 2) родной веб-интерфейс доступа к данным (без реализации бизнес-логики);
- 3) подход к представлению, хранению, доступу к данным.

Данные лежат в множестве реляционных баз, библиотека `lib_meta.jar` (классы пакета `arpt.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 также 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. ^[3]

Далее будут рассмотрены задания, предоставленные ООО «Комплексные системы».

1.3 Анализ поставленной задачи

Данные в МЕТЕ представлены в виде совокупности объектов определенных типов. Каждый тип имеет идентификатор из 3 цифр. Каждый объект имеет идентификатор из 12 цифр, включая 3 цифры идентификатора типа и единицу перед ними. Объекты имеют некоторые атрибуты – их называют атрибутами типа. Атрибуты типа также имеют свои собственные идентификаторы из 10 цифр, которые включают 3 цифры идентификатора типа, единицу перед ними и кодовую цифру типа (число, строка, дата и т.д.).^[3]

Для выполнения заданий студентам-практикантам предоставляется тестовый сервер, на котором уже имеются некоторые типы, атрибуты типов и объекты.

Ниже представлена постановка задач компании:

- 1) Сделать форму в HTML для создания описания экскурсии (тип 506):
 - название (идентификатор атрибута типа 1506410000);
 - описание краткое (идентификатор атрибута типа 1506410282);
 - регионы (идентификатор атрибута типа 1506923461);
 - дополнительно оплачиваются входные билеты (идентификатор атрибута типа 1506223120);
 - бронировать у партнера (идентификатор атрибута типа 1506910189) - ссылка на партнера 158 типа, название (идентификатор атрибута типа 1001211);
 - Тип (кодированные обозначения: экскурсия - 0, билет - 1, спорт - 2, прокат - 3, услуга - 4, СПА - 5, авиация - 6) (идентификатор атрибута типа 1506310181).
- 2) Сделать возможность удалять объекты 506 типа в форме - заносим ID объекта, который надо удалить, по кнопке «Удалить» - объект удаляется.
- 3) Написать интерфейс, чтобы отображать данные из Redis. Зашли в интерфейс, написать откуда взяли данные, если в редисе нет ключа, написать, что заново положили ключ+значение (тип 46) в Redis и положить этот ключ+значение, или просто достали из редиса - написали об этом. В интерфейсе по ID 46 надо

отображать следующую информацию:

- Название (идентификатор атрибута типа 1000348);
- ID номера (идентификатор атрибута типа 1000350);
- тип стоимости (0/1/2 - С/БНС/НС) (идентификатор атрибута типа 1046222729).

Все кладем в Redis «rev», время жизни ключа 3 минуты. (идентификаторы объектов для теста 104610001184, 104610000865, 104610000863, 104610000831, 104610000807, 104610000783, 104610000561, 104610000529, 104610000495).

Такова трактовка заданий, предоставленная ООО «Комплексные системы».

Вывод, анализ поставленной задачи выполнен. Далее будет изложена разработка приложения для работы с системой МЕТА в базах данных PostgreSQL и Redis.

2 Технология реализации

В данном разделе будет показан код сервлета на языке Java, который, в зависимости от GET-параметров, будет выполнять запросы к системе META, затем заворачивать их в требуемый формат – это HTML. Прежде всего необходимо прописать индексацию сервлета.

2.1 Настройка индексации сервлета

Подобные настройки должны быть расположены в файле web.xml, ниже представлен значащий отрывок из этого файла – его полная часть изложена в Приложении А (листинг 2.1).

Листинг 2.1 – отрывок из Приложения А, файл web.xml

```
<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>
```

Здесь указано имя сервлета, класс сервлета и адрес страницы, по которому можно обратиться в браузере и который должен быть дополнен адресом хоста.

Таким образом, был проиндексирован мой сервлет и при обращении к серверу с запросом веб-страницы “/eldar”, будет выводиться результат работы написанного мной сервлета EldarServlet, оформление которого, а также дизайн веб-интерфейса, будут приведены в следующем подразделе. ^[4]

2.2 Оформление кода и дизайн веб-интерфейса

Для ориентации в коде необходима нумерация строк, поэтому код будет прилагаться скриншотами. Изложение пакета и подключения сторонних пакетов библиотек имеет смысл опустить. Ниже приведено определение класса-сервлета и

показаны поля экземпляра, статические поля и статическое определение. Полный код класса находится в Приложении Б (листинг 2.2).

Листинг 2.2 – Объявление класса-сервлета

```
public class EldarServlet extends HttpServlet {
    private PrintWriter out;
    private ResourceBundle mains;
    private HttpServletRequest request;
    private HttpServletResponse response;
    private final int myId = 1000360;
    private final String redis = "rev";
    private final String prefix = "Eldar";

    private final Map<String, String> typeMap = new HashMap<>();
    private final Map<String, String> yesnoMap = new HashMap<>();
    private final Map<String, List<String>> pagesMap = new TreeMap<>();
    private List<Procedure> solutionList;
```

В коде объявлены объект потока вывода, обращения к ресурсам, заголовков запроса, заголовков ответа, мой личный идентификатор для работы с системой МЕТА, имя сервера Redis, мой префикс для ключей в Redis, затем словари для типов мероприятий, для утвердительных ответов и для навигации по веб-интерфейсу.

Далее будет представлен метод обработки GET-запросов (листинг 2.3).

Листинг 2.3 – Методы обработки GET-запросов

```
public void doGet(HttpServletRequest req, HttpServletResponse res) throws IOException {
    initialize(req, res);
    try {
        int task = Integer.parseInt(request.getParameter("task"));
        if(task < 1 || task > solutionList.size()) {
            throw new NumberFormatException();
        }
        solutionList.get(task - 1).run();
    } catch (NumberFormatException e) {
        printHtml("Hello", "<h1 style=\"text-align:center;\">Привет Sirius!</h1>");
    } catch (Exception e) {
        printException(e);
    }
    out.flush();
}
```

Здесь выполняется метод инициализации и выполняется метод задания, соответствующий номеру задания из запроса. Аналогично, ниже будет приведен метод обработки POST-запросов (листинг 2.4).^[4]

Листинг 2.4 – Метод обработки POST-запросов

```
public void doPost(HttpServletRequest req, HttpServletResponse res) throws IOException {
    initialize(req, res);
    String task = request.getParameter("task");
    try {
        if ("7".equals(task)) {
            lection3task7post();
        }
        if ("13".equals(task)) {
            lection5task1post();
        }
    } catch (Exception e) {
        printException(e);
    }
    out.flush();
}
```

Здесь также выполняется метод инициализации и выполняется метод задания, однако использование метода запросов POST используется только в 7 задании – здесь это 2 задание, оно будет показано позже. Далее представлен код инициализации (листинг 2.5).

Листинг 2.5 – Инициализация

```
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"));

    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");

    pagesMap.put("PostgreSQL", Arrays.asList("Один", "Два",
        "Три", "Четыре", "Пять", "Шесть", "Семь", "Восемь"));
    pagesMap.put("Redis", Arrays.asList("Один", "Два", "Три", "Четыре"));
    pagesMap.put("Авторизация", Arrays.asList("Один", "Два", "Три", "Четыре", "Пять"));

    solutionList = Arrays.asList(this::lection3task1, this::lection3task2, this::lection3task3,
        this::lection3task4, this::lection3task5, this::lection3task6, this::lection3task7,
```

```

        this::lection3task8, this::lection4task1, this::lection4task2, this::lection4task3,
        this::lection4task4, this::lection5task1, this::lection5task2, this::lection5task3,
        this::lection5task4, this::lection5task5, this::cheatCode, this::cheatFunction);
    }

```

Суть данного метода заключается в присвоении ссылок на объекты заголовков запроса и ответа в поля экземпляра сервлета, а также уточнение кодировки, получение ссылки на объект потока вывода и объект обращения к ресурсам. В ходе выполнения программы могут возникать исключения, которые обрабатываются в методах `doGet` и `doPost`, вывод текста исключений показан ниже (листинг 2.6).

Листинг 2.6 – Вывод текста исключений

```

public void printException(Exception e) {
    StackTraceElement[] stack = e.getStackTrace();
    String[] stackTrace = new String[stack.length];
    for(int i = 0; i < stack.length; ++i) {
        stackTrace[i] = stack[i].toString();
    }
    printHtml("Exception!", String.format("<h1>%s: %s</h1><div>%s</div>",
        e.getClass().getName(), e.getMessage(), String.join("<br>", stackTrace)));
}

```

Каждое задание и вывод исключений формируют определенный текст, помимо это каждая страница должна иметь название и стили — всем этим пользуется метод вывод HTML кода, который приведен ниже (листинг 2.7).

Рисунок 2.7 – Вывод HTML-кода

```

public void printHtml(String title, String text) {
    response.setContentType("text/html; charset=UTF-8");
    String task = request.getParameter("task");
    StringBuilder output = new StringBuilder();
    output.append(String.format("<!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; }" +
        "ul { list-style: none; }" +
        "ul li:before { content: \"—\"; position: relative; left: -5px; }" +
        "</style></head>" +

```

```

        "<body><div style=\"text-align:center;margin-bottom:20px;\">" +
        "<table class=\"form\">", title));
int tasks = 0;
for(String key : pagesMap.keySet()) {
    output.append(String.format("<tr>" +
        "<td style=\"font-weight:bold;color:#072d78;padding-right:15px;\">" +
        "%s</td><td>", key));
    List<String> numbers = pagesMap.get(key);
    for(int i = 1, n = numbers.size(); i <= n; ++i) {
        if(String.valueOf(tasks + i).equals(task)) {
            output.append(String.format("<b style=\"color:grey\">%s</b>%s",
                numbers.get(i - 1), i < n ? " | " : ""));
        } else {
            output.append(String.format("<a href=\"?task=%d\">%s</a>%s",
                tasks + i, numbers.get(i - 1), i < n ? " | " : ""));
        }
    }
    output.append("</td></tr>");
    tasks += numbers.size();
}
output.append(String.format("</table></div>%s</body></html>", text));
out.print(output);
}

```

По итогу получаем приложенный ниже веб-интерфейс. Имеются две секции заданий: система META в PostgreSQL и в Redis, - а далее ссылки на сами задания, ниже вывод самого задания. В данном случае, шестое задание, которое не будет рассматриваться в отчете, но очень демонстративно в плане дизайна – есть и форма, и таблица – элементы, на которые был упор в стилях CSS (рис. 2.1).

[PostgreSQL](#)
[Один](#) | [Два](#) | [Три](#) | [Четыре](#) | [Пять](#) | [Шесть](#) | [Семь](#) | [Восемь](#)
[Redis](#)
[Один](#) | [Два](#) | [Три](#) | [Четыре](#)

Тип
 Код
 Адрес
 Категория
 Контрагент

№	Название	Тип	Адрес	Да/Нет	Контрагент
1	(Переименовал) Тест 506/36 - 10	Концерт	Олимпийский проспект, д.1	-	КонтрАг
2	(Переименовал) Тест 506/36 - 20	Концерт	Олимпийский проспект, д.1	-	КонтрАг
3	(Переименовал) Тест 506/36 - 240	Концерт	Олимпийский проспект, д.1	-	КонтрАг
4	(Переименовал) Тест 506/36 - 30	Концерт	Олимпийский проспект, д.1	-	КонтрАг
5	(Переименовал) Тест 506/36 - 360	Концерт	Олимпийский проспект, д.1	-	КонтрАг
6	(Переименовал) Тест 506/36 - 40	Концерт	Олимпийский проспект, д.1	-	КонтрАг
7	(Переименовал) Тест 506/36 - Базис	Концерт	Олимпийский проспект, д.1	-	КонтрАг

Рисунок 2.1 – Вывод шестого задания в браузере

Таким образом, было показано оформление Java-кода, показаны стили CSS и пример вывода HTML-кода с его визуализацией в браузере. В следующем подразделе будут рассматриваться задания.

2.3 Выполнение запросов к системе МЕТА

Первым на очереди будет первое задание, в котором демонстрируется выборка данных из базы – в данном случае, выборка областей, а затем соответствующих выбранной области городов.

Далее будет представлено задание с созданием нового объекта системы МЕТА и его последующая загрузка в базу (листинг 2.8).

Листинг 2.8 – Гипертекст формы создания объекта

```
public void lection3task7() 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\">");
    List<String> regionIds = intern.Utils.getKeysSortedByValue(regions, false);
    for(String regionId : regionIds) {
        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);
}
```

Суть показанного метода в выводе формы в браузер, а также выводе всех объектов данного типа, реализованном в другом методе. Результат работы метода можно наблюдать ниже (рис. 2.2).

Название: 1 сентябре в Политехе

Описание: Тестовое описание

Регион: Пермь

Доп. оплата: 100

Бронирование у партнера: тестовый Сириус

Тип: Концерт

Создать

Рисунок 2.2 – Форма создания объекта

Здесь показана форма, дополнительно заполненная для создания и записи объекта в систему МЕТА. Результатом нажатия на кнопку «Создать» будет дальнейший вывод в таблице нового объекта (рис. 2.3).

№	ID	Название	Описание	Регион	Доп. оплаты	Бронирование у партнера	Тип
1	150610003700	1 сентябре в Политехе	Тестовое описание	Пермь	100	тестовый Сириус	Концерт
2	150610003660	Агрызок	Агрыз - город мечты	null	1	тестовый Сириус	Билет
3	150610003560	title	description	null	0	тестовый Сириус	Экскурсия

Рисунок 2.3 – Новый объект

Теперь же следует перейти к рассмотрению того, как происходит обработка POST-запроса, создание объекта и его загрузка в систему МЕТА. Данный код представлен ниже (листинг 2.9).

Листинг 2.9 – Создание объекта и его загрузка в базу

```
public void lection3task7post() 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");
}

```

Здесь создается объект, получает свои атрибуты из POST-запроса, отправляется в базу данных и выполняется перенаправление с GET-запросом на страницу с заданием. Далее будут показано извлечение из базы регионов и партнеров, необходимых для выполнения задания (листинг 2.10).

Листинг 2.10 – Извлечение регионов и партнеров

```

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"));
    }
}

```

Как видно, все очень просто и не требует детального рассмотрения. Теперь же следует приступить к выполнению следующего задания, код которого представлен ниже (листинг 2.11).

Листинг 2.11 – Удаление объекта из системы МЕТА

```
public void lection3task8() 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);
}
```

Выполнение данного кода выведет в браузер форму для удаления (рис. 2.4).



Рисунок 2.4 – Форма удаления

Далее будет представлено задание с загрузкой объекта системы МЕТА в базу данных на оперативной памяти Redis (листинг 2.12).

Листинг 2.12 – Загрузка объекта в Redis

```
public void lection4task1() throws Exception {
    long time = 1000000000L;
    int expire = 180;
    String[] types = new String[]{"C", "БНС", "НС"};
    String costId = Util.s2s(request.getParameter("id"));
    String foundOrCreated = null;
    Obb ob = null;
    if(!costId.equals("")) {
        ob = Ob0.fromBytes(Ob3.get(mains, redis, redis, (prefix + costId).getBytes()));
        if(ob == null) {
            ob = Ob0.getOb(mains, costId);
            if(ob == null) {
```



```

        foundOrCreated = "<div style=\"text-align:center;color:red;\">" +
            "Искомый объект не существует в базе PostgreSQL</div>";
    } else {
        Ob3.puts(mains, redis, (prefix + costId).getBytes(), Ob0.toBytes(ob),
            time, false, expire, "");
        foundOrCreated = "<div style=\"text-align:center;color:blue;\">" +
            "Объект изъят из базы PostgreSQL и записан в REDIS</div>";
    }
} else {
    foundOrCreated = "<div style=\"text-align:center;color:green;\">Объект найден</div>";
}
}
String text = String.format("<form method=\"get\" action=\"#\>" +
    "ID <input type=\"text\" name=\"id\" value=\"%s\"> " +
    "<input type=\"submit\" value=\"Найти\"> " +
    "<input type=\"hidden\" name=\"task\" value=\"9\"> " +
    "</form>%s%s", costId, foundOrCreated == null ? "" : foundOrCreated, ob == null ? "" :
    String.format("<table class=\"data\">" +
        "<tr><td>Название</td><td>ID номера</td><td>Тип стоимости</td></tr>" +
        "<tr><td>%s</td><td>%s</td><td>%s</td></tr>" +
        "</table>",
        ob.getAt(1000348),
        ob.getAt(1000350),
        types[Integer.parseInt(ob.getAt(1046222729))]);
printHtml("Redis - Один", text);
}

```

Вывод задания представляет собой форму с возможностью ввода ID. Если его ввести нажать на кнопку «Найти», будет выведены данные найденного объекта в таблице (рис. 2.5).

ID

Объект изъят из базы PostgreSQL и записан в REDIS

Название	ID номера	Тип стоимости
СТАНДАРТ 2-Х МЕСТН. ВВ	102110001662	НС

Рисунок 2.5 – Форма и таблицы с данными найденного объекта

Вывод, в данном разделе была показана реализация поставленной задачи, а именно, реализация приложения для работы с системой META в PostgreSQL и Redis.

Заключение

Предоставляя место для прохождения практики, ООО «Комплексные системы» ставило перед собой цель подготовить для себя будущих специалистов. С подачи компании был изучен весь стек технологий, применяемых на производстве собственного программного обеспечения.

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 системой МЕТА выполнена.

Список использованных источников

- 1) Группа компаний Комплексные системы [Электрон. ресурс]. URL: <https://www.complex-systems.biz/>
- 2) О системе МЕТА [Электрон. ресурс]. URL: <https://www.complex-systems.biz/meta.html>
- 3) Документация по системе МЕТА [Электрон. ресурс]. URL: <https://sirius-systems.ru/documentation.html>
- 4) Шилдт, Герберт. Java. Полное руководство, 10-е изд. : Пер. с англ. – СПб. : ООО «Диалектика», 2020. – 1488 с. : ил. – Парал. тит. англ.

Приложение А

Листинг А – файл web.xml

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

    <servlet>
        <servlet-name>InokovaServlet</servlet-name>
        <servlet-class>intern.InokovaServlet</servlet-class>
        <init-param>
            <param-name>main_resource</param-name>
            <param-value>intern_main</param-value>
        </init-param>
    </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>

    <servlet>
        <servlet-name>ServletKartashova</servlet-name>
        <servlet-class>intern.ServletKartashova</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>ServletKartashova</servlet-name>
        <url-pattern>/kartashova</url-pattern>
    </servlet-mapping>

</web-app>
```

Приложение Б

Листинг Б – файл EldarServlet.java

```
package intern;

import appt.meta3.*;
import appt.meta3.servlet.AuthServlet;
import org.apache.commons.lang3.math.NumberUtils;

import javax.servlet.http.Cookie;
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.nio.charset.StandardCharsets;
import java.text.*;
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 final String redis = "rev";
    private final String prefix = "Eldar";

    private final Map<String, String> typeMap = new HashMap<>();
    private final Map<String, String> yesnoMap = new HashMap<>();
    private final Map<String, List<String>> pagesMap = new TreeMap<>();
    private List<Procedure> solutionList;

    public interface Procedure {
        void run() throws Exception;
    }

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws IOException {
        initialize(req, res);
        String task = request.getParameter("task");
        try {
            if ("7".equals(task)) {
                lection3task7post();
            }
            if ("13".equals(task)) {
                lection5task1post();
            }
        } catch (Exception e) {
            printException(e);
        }
        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 > solutionList.size()) {
            throw new NumberFormatException();
        }
        solutionList.get(task - 1).run();
    } catch (NumberFormatException e) {
        printHtml("Hello", "<h1 style='text-align:center;'>Привет Sirius!</h1>");
    } catch (Exception e) {
        printException(e);
    }
    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"));

    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");

    pagesMap.put("PostgreSQL", Arrays.asList("Один", "Два",
        "Три", "Четыре", "Пять", "Шесть", "Семь", "Восемь"));
    pagesMap.put("Redis", Arrays.asList("Один", "Два", "Три", "Четыре"));
    pagesMap.put("Авторизация", Arrays.asList("Один", "Два", "Три", "Четыре", "Пять"));

    solutionList = Arrays.asList(this::lection3task1, this::lection3task2, this::lection3task3,
        this::lection3task4, this::lection3task5, this::lection3task6, this::lection3task7,
        this::lection3task8, this::lection4task1, this::lection4task2, this::lection4task3,
        this::lection4task4, this::lection5task1, this::lection5task2, this::lection5task3,
        this::lection5task4, this::lection5task5, this::cheatCode, this::cheatFunction);
}

public void printException(Exception e) {
    StackTraceElement[] stack = e.getStackTrace();
    String[] stackTrace = new String[stack.length];
    for(int i = 0; i < stack.length; ++i) {
        stackTrace[i] = stack[i].toString();
    }
    printHtml("Exception!", String.format("<h1>%s: %s</h1><div>%s</div>",
        e.getClass().getName(), e.getMessage(), String.join("<br>", stackTrace)));
}

public void lection3task1() throws Exception {
    Map<String, String> regions = getRegions("100410000050");
    Obb filter = Obb.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);
    List<String> regionIds = intern.Utills.getKeysSortedByValue(regions, false);
    StringBuilder text = new StringBuilder("<form action=\"#\" method=\"get\">" +
        "<table class=\"form\"><tr><td>Регион</td><td><select name=\"region\">" +
    for(String key : regionIds) {
        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>");
    Arrays.sort(cities, Comparator.comparing((Obb ob) -> ob.getAt("1000098")));
    int i = 0;
    text.append("<table class=\"data\"><tr><td>№</td><td>ID</td><td>Город</td></tr>");
    for(Obb city : cities) {
        text.append(String.format("<tr><td>%d</td><td>%s</td><td>%s</td></tr>",
            ++i, city.id, 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 lection3task2() 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);
    List<String> regionIds = intern.Utills.getKeysSortedByValue(regions, false);
    Arrays.sort(countries, Comparator.comparing((Obb ob) -> ob.getAt("1000000")));
    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>");
if(regionIds.isEmpty()) {
    text.append("<h4 style=\"text-align:center;color:grey;\">" +
        "Данная страна не поддерживает разбиение на регионы</h4>");
} else {
    text.append("<table class=\"data\"><tr><td>№</td><td>Область</td><td>Города</td></tr>");
    int i = 0;
    for (String key : regionIds) {
        if (regions.containsKey(key) && distribution.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 lection3task3() 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);
    Arrays.sort(tourists, Comparator.comparing(
        (Obb ob) -> ob == null ? "" : ob.getAt("1000144")));

    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 lection3task4() 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);
    Arrays.sort(costs, Comparator.comparing((Obb ob) -> ob.getAt("1036423021")));
    Arrays.sort(agents, Comparator.comparing((Obb ob) -> ob.getAt("1317100000")));
    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 lection3task5() 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 lection3task6() 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);
    Arrays.sort(agents, Comparator.comparing((Obb ob) -> ob.getAt("1317100000")));

    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 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>Название</td><td>Тип</td>" +
    "<td>Адрес</td><td>Да/Нет</td><td>Контрагент</td></tr>");
Arrays.sort(costs, Comparator.comparing((Obb ob) -> ob.getAt("1036423021")));
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, cost.getAt("1036423021"),
        typeMap.get(cost.getAt("1036200042")),
        cost.getAt("1036410028"),
        yesnoMap.get(cost.getAt("1162200125")),
        agentMap.get(cost.getAt("1036922797"))));
}
printHtml("Экскурсии", text.append("</table>").toString());
}

public void lection3task7() 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\">");
    List<String> regionIds = intern.Utils.getKeysSortedByValue(regions, false);
    for(String regionId : regionIds) {
        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 lection3task7post() 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 lection3task8() 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[] desc = 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 : desc) {
        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 lection4task1() throws Exception {
    long time = 100000000L;
    int expire = 180;
    String[] types = new String[]{"C", "БНС", "НС"};
    String costId = Util.s2s(request.getParameter("id"));
    String foundOrCreated = null;
    Obb ob = null;
    if(!costId.equals("")) {
        ob = Ob0.fromBytes(Ob3.get(mains, redis, redis, (prefix + costId).getBytes()));
        if(ob == null) {
            ob = Ob0.getOb(mains, costId);
            if(ob == null) {
                foundOrCreated = "<div style=\"text-align:center;color:red;\">" +
                    "Искомый объект не существует в базе PostgreSQL</div>";
            } else {
                Ob3.puts(mains, redis, (prefix + costId).getBytes(), Ob0.toBytes(ob),
                    time, false, expire, "");
                foundOrCreated = "<div style=\"text-align:center;color:blue;\">" +
                    "Объект изъят из базы PostgreSQL и записан в REDIS</div>";
            }
        } else {
            foundOrCreated = "<div style=\"text-align:center;color:green;\">Объект найден</div>";
        }
    }
    String text = String.format("<form method=\"get\" action=\"%#\"> " +
        "ID <input type=\"text\" name=\"id\" value=\"%s\"> " +
        "<input type=\"submit\" value=\"Найти\"> " +
        "<input type=\"hidden\" name=\"task\" value=\"9\"> " +
        "</form>%s%s", costId, foundOrCreated == null ? "" : foundOrCreated, ob == null ? "" :
        String.format("<table class=\"data\"> " +
            "<tr><td>Название</td><td>ID номера</td><td>Тип стоимости</td></tr> " +
            "<tr><td>%s</td><td>%s</td><td>%s</td></tr> " +
            "</table>",
            ob.getAt(1000348),
            ob.getAt(1000350),

```

```

        types[Integer.parseInt(ob.getAt(1046222729))]);
    printHtml("Redis - Один", text);
}

public void lection4task2() throws Exception {
    long time = 100000000L;
    int expire = 1;
    String action = request.getParameter("action");
    action = action == null || action.equals("") ? "find" : action;
    String key = Util.s2s(request.getParameter("key"));
    String value = null;
    if(action.equals("find") && !key.equals("")) {
        byte[] bytes = Ob3.get(mains, redis, redis, (prefix + key).getBytes());
        value = bytes == null ? "" : new String(bytes, StandardCharsets.UTF_8);
    } else if (action.equals("create")) {
        value = request.getParameter("value");
        Ob3.puts(mains, redis, (prefix + key).getBytes(), value.getBytes());
    } else if (action.equals("delete")) {
        //Ob3.del(mains, redis, prefix + key, redis);
        byte[] bytes = Ob3.get(mains, redis, redis, (prefix + key).getBytes());
        value = bytes == null ? "" : new String(bytes, StandardCharsets.UTF_8);
        Ob3.puts(mains, redis, (prefix + key).getBytes(), "", getBytes(),
            time, false, expire, "");
    }
    String text = String.format("<form method=\"get\" action=\"#\"><table class=\"form\">\" +
        "<tr><td>Ключ</td><td><input type=\"text\" name=\"key\"></td></tr>\" +
        "<tr><td>Значение</td><td><input type=\"text\" name=\"value\"></td></tr>\" +
        "</table>\" +
        "<input type=\"submit\" value=\"Создать\">\" +
        "<input type=\"hidden\" name=\"task\" value=\"10\">\" +
        "<input type=\"hidden\" name=\"action\" value=\"create\">\" +
        "</form>%s\" +
        "<form method=\"get\" action=\"#\">\" +
        "Ключ <input type=\"text\" name=\"key\">\" +
        "<input type=\"submit\" value=\"Найти\">\" +
        "<input type=\"hidden\" name=\"task\" value=\"10\">\" +
        "<input type=\"hidden\" name=\"action\" value=\"find\">\" +
        "</form>%s\" +
        "<form method=\"get\" action=\"#\">\" +
        "Ключ <input type=\"text\" name=\"key\">\" +
        "<input type=\"submit\" value=\"Удалить\">\" +
        "<input type=\"hidden\" name=\"task\" value=\"10\">\" +
        "<input type=\"hidden\" name=\"action\" value=\"delete\">\" +
        "</form>%s\",
        action.equals("create") ? "<p style=\"text-align:center;color:green\">[" +
            key + " : " + value + "]" создано</p> : "<br>",
        !key.equals("") && action.equals("find") ?
            !value.equals("")
                ? "<p style=\"text-align:center;color:green\">[" +
                    key + " : " + value + "]"</p>"
                : "<p style=\"text-align:center;color:red\">Объект с ключом " +
                    key + " не найден.</p>"
            : "<br>",
        action.equals("delete")
            ? !value.equals("")
                ? "<p style=\"text-align:center;color:green\">[" +
                    key + "]" удален</p>"
                : "<p style=\"text-align:center;color:red\">Объект с ключом " +
                    key + " не найден.</p>"
            : "<br>");
    printHtml("Redis - Два", text);
}

```



```

}

public void lection4task3() throws Exception {
    Obb[] obs = Ob0.getSrcObs(mains, Ob0.createFilter(36), 0, 0);
    List<String> result = new ArrayList<>(obs.length);
    String action = Util.s2s(request.getParameter("action"));
    action = action.equals("") ? "postgre" : action;
    long time = -1;
    if(action.equals("postgre")) {
        time = catchTime() -> {
            for(Obb ob : obs) {
                result.add(Ob0.getOb(mains, ob.id).id);
            }
        };
    } else if(action.equals("redis")) {
        time = catchTime() -> {
            for(Obb ob : obs) {
                Obb temp = Ob0.fromBytes(Ob3.get(mains, redis, redis, (prefix + ob.id).getBytes()));
                if(temp != null) {
                    result.add(temp.id);
                } else {
                    result.clear();
                    break;
                }
            }
        };
    } else if(action.equals("in_redis")) {
        long time0 = 1000000000L;
        int expire = 3600;
        for(Obb ob : obs) {
            Ob3.puts(mains, redis, (prefix + ob.id).getBytes(), Ob0.toBytes(ob),
                time0, false, expire, "");
        }
    } else {
        throw new Exception("Плохой аргумент");
    }
    StringBuilder text = new StringBuilder();
    text.append("<div style=\"text-align:center;\"><p> +
        "<a href=\"?task=11&action=postgre\">Выбрать из PostgreSQL</a> | " +
        "<a href=\"?task=11&action=redis\">Выбрать из Redis</a> | " +
        "<a href=\"?task=11&action=in_redis\">Заполнить Redis</a></p>");
    text.append(time == -1
        ? String.format("<p style=\"color:green\">%d объектов добавлено в Redis.</p>",
            obs.length)
        : !result.isEmpty()
        ? String.format("<p>Время выборки из %s: %d мс</p>",
            action.equals("postgre") ? "PostgreSQL" : "Redis", time)
        : "<h1 style=\"text-align:center;\"> +
        "<a href=\"?task=11&action=in_redis\">Заполните</a> Redis</h1>");
    text.append("</div>");
    if(!result.isEmpty()) {
        text.append(String.join(", ", result));
    }
    printHtml("Redis - Три", text.toString());
}

public long catchTime(Procedure procedure) throws Exception {
    long time = System.currentTimeMillis();
    procedure.run();
    return System.currentTimeMillis() - time;
}

```

```

public void lection4task4() throws Exception {
    printHtml("Redis - Четыре",
        "<div style=\"border:1px solid #ccc; width:900px; margin:0px auto; padding:15px;\">" +
        "Целесообразно применять Redis в: <ul>" +
        "<li>онлайн-магазинах для корзины;" +
        "<li>онлайн-играх по типу шахмат для хранения ходов и состояния шахматной доски;" +
        "<li>программах с видеоконференциями для буферизации видеопотоков;" +
        "<li>в общем, применима в тех случаях, когда требуется быстроедействие и безопасна
возможная" +
        "потеря данных в связи с отключением электричества на сервере." +
        "</ul></div>");
}

public void lection5task1() throws Exception {
    String userEmail = getUserEmail();
    String tourId = request.getParameter("tour_id");
    String quoteId = request.getParameter("quote_id");
    StringBuilder text = new StringBuilder();
    new StackPager()
        .add() -> true,
        () -> "task=13",
        () -> "Экскурсии",
        (String link) -> toursHtml(text, userEmail))
        .add() -> NumberUtils.isNumber(tourId),
        () -> "tour_id=" + tourId,
        () -> "Квоты экскурсии ID " + tourId,
        (String link) -> quotesHtml(text, tourId))
        .add() -> NumberUtils.isNumber(quoteId),
        () -> "quote_id=" + quoteId,
        () -> "Квота ID " + quoteId,
        (String link) -> quoteEditHtml(text, quoteId))
        .run(text);
    printHtmlWithAuth(userEmail, "Авторизация - Один", text.toString());
}

public void toursHtml(StringBuilder text, String userEmail) throws Exception {
    Obb tourFilter = Obb.createFilter(36);
    Obb quoteFilter = Obb.createFilter(799);
    Obb[] tours = Obb.getSrcObs(mains, tourFilter, 0, 0);
    Obb[] quotes = Obb.getSrcObs(mains, quoteFilter, 0, 0);
    Set<String> quotedTours = new HashSet<>();
    for(Obb quote : quotes) {
        quotedTours.add(quote.getAt(1799910177));
    }
    text.append("<table class=\"data\"><tr><td>№</td><td>Код экскурсии</td><td>Адрес</td></tr>");
    int i = 0;
    for(Obb tour : tours) {
        text.append(String.format("<tr><td>%d</td><td>%s</td><td>%s</td></tr>",
            ++i,
            quotedTours.contains(tour.id) && userEmail != null
                ? "<a href=\"?task=13&tour_id=" + tour.id + "\">" + tour.id + "</a>"
                : tour.id,
            tour.getAt("1036423021") + "<br>" + tour.getAt("1036410028")));
    }
    text.append("</table>");
}

public void quotesHtml(StringBuilder text, String tourId) throws Exception {
    Obb filter = Obb.createFilter(799);
    Obb.addCondition(filter, 1799910177, Obb.ComparisonType.EQ, tourId);

```



```

Obb[] quotes = Ob0.getSrcObs(mains, filter, 0, 0);
text.append("<table
class=\"data\"><tr><td>№</td><td>ID</td><td>Дата</td><td>Количество</td></tr>");
int i = 0;
for(Obb quote : quotes) {
    text.append(String.format("<tr><td>%d</td><td>%s</td><td>%s</td><td>%s</td></tr>", ++i,
        String.format("<a href=\"?task=13&tour_id=%s&quote_id=%s\">%s</a>", tourId, quote.id, quote.id),
        quote.getAt("1799510027"), quote.getAt("1799210047")));
}
text.append("</table>");
}

public void quoteEditHtml(StringBuilder text, String quoteId) throws Exception {
    Obb quote = Ob0.getOb(mains, quoteId);
    text.append(String.format("<form method=\"post\" action=\"#\"> +
        "<table class=\"form\"> +
        "<tr><td>Дата</td><td><input type=\"text\" name=\"date\" value=\"%s\"></td></tr> +
        "<tr><td>Количество</td><td><input type=\"text\" name=\"count\" value=\"%s\"></td></tr> +
        "</table><input type=\"hidden\" name=\"task\" value=\"13\"> +
        "<input type=\"hidden\" name=\"tour_id\" value=\"%s\"> +
        "<input type=\"hidden\" name=\"quote_id\" value=\"%s\"> +
        "<input type=\"submit\" value=\"Обновить квоту\"></form>",
        quote.getAt("1799510027"), quote.getAt("1799210047"),
        quote.getAt("1799910177"), quote.id));
}

public void lection5task1post() throws Exception {
    String tourId = request.getParameter("tour_id");
    String quoteId = request.getParameter("quote_id");
    String date = request.getParameter("date");
    String count = request.getParameter("count");
    Obb quote = Ob0.getOb(mains, quoteId);
    Ob0.addAt(quote, "1799510027", date);
    Ob0.addAt(quote, "1799210047", count);
    quote.id_user = myId;
    Ob0.edtOb(mains, quote);
    response.sendRedirect("eldar?task=13&tour_id=" + tourId + "&quote_id=" + quoteId);
}

public void lection5task2() throws Exception {
    String userEmail = getUserEMail();
    String action = request.getParameter("action");
    String quoteId = request.getParameter("id");
    StringBuilder text = new StringBuilder();
    if(quoteId != null) {
        Obb quote = Ob0.getOb(mains, quoteId);
        DateFormat format = new SimpleDateFormat("dd.MM.yyyy");
        Date date = format.parse(quote.getAt(1074100143));
        if(date.after(new Date())) {
            if("add".equals(action)) {
                Ob0.addAt(quote, 1074200033, "1");
                text.append("<div style=\"text-align:center;color:green;\">Стоп добавлен</div>");
            } else if("del".equals(action)) {
                Ob0.addAt(quote, 1074200033, "0");
                text.append("<div style=\"text-align:center;color:green;\">Стоп убран</div>");
            }
            Ob0.edtOb(mains, quote);
        } else {
            text.append("<div style=\"text-align:center;color:red;\">Стоп не добавлен</div>");
        }
    }
}

```

```

Obb filter = Ob0.createFilter(74);
Obb[] quotes = Ob0.getSrcObs(mains, filter, 0, 0);
text.append("<table class=\"data\">");
text.append(String.format("<tr><td>№</td><td>ОТЕЛЬ</td><td>Страна</td> " +
    "<td>Дата начала</td><td>Дата окончания</td><td>Строп</td>%s</tr>",
    userEmail == null ? "" : "<td>Действие</td>"));
int i = 0;
for(Obb quote : quotes) {

text.append(String.format("<tr><td>%d</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td>%
s</tr>",
    ++i, Ob0.getZn(mains, quote.getAt(1000538), 1000127, 4),
    Ob0.getZn(mains, quote.getAt(1000802), 1000000, 4),
    quote.getAt(1074100142),
    quote.getAt(1074100143),
    quote.getAt(1074200033),
    userEmail == null ? "" : "<td>" + (quote.getAt(1074200033).equals("0")
        ? "<a href=?task=14&action=add&id=" + quote.id + "\">Добавить</a>"
        : "<a href=?task=14&action=del&id=" + quote.id + "\">Удалить</a>") + "</td>"));
}
text.append("</table>");
printHtmlWithAuth(userEmail, "Авторизация - Два", text.toString());
}

public void lection5task3() throws Exception {
    String userEmail = getUserEmail();
    StringBuilder text = new StringBuilder();
    text.append(hasRole("1001900012") + "<br>");
    text.append(hasTitle("1007410000") + "<br>");
    printHtmlWithAuth(userEmail, "Авторизация - Три", text.toString());
}

public void lection5task4() throws Exception {
    String userEmail = getUserEmail();
    String text;
    if(userEmail == null) {
        text = "<div style=\"text-align:center;\">Вы не авторизированы.</div>";
    } else {
        User whoami = AuthServlet.isAuth(request, response, mains, "meta");
        text = String.format("<table class=\"form\"> " +
            "<tr><td>ID: </td><td>%s</td></tr> " +
            "<tr><td>Тип ID: </td><td>%d</td></tr> " +
            "<tr><td>Статус: </td><td>%s</td></tr> " +
            "<tr><td>Логин: </td><td>%s</td></tr> " +
            "<tr><td>Почта: </td><td>%s</td></tr> " +
            "<tr><td>IP: </td><td>%s</td></tr> " +
            "<tr><td>User-Agent: </td><td>%s</td></tr></table>",
            whoami.id, Ob0.id2type(whoami.id), whoami.status, whoami.login, whoami.mail,
            request.getHeader("X-Real-IP"), request.getHeader("User-Agent"));
    }
    printHtmlWithAuth(userEmail, "Авторизация - Четыре", text);
}

public void lection5task5() throws Exception {
    String userEmail = getUserEmail();
    String countryId = request.getParameter("country_id");
    String regionId = request.getParameter("region_id");
    String cityId = request.getParameter("city_id");
    String hotelId = request.getParameter("hotel_id");
    String roomId = request.getParameter("room_id");
    String nsId = request.getParameter("ns_id");

```

```

String quoteId = request.getParameter("quote_id");
StringBuilder text = new StringBuilder();
new StackPager()
    .add() -> true,
        () -> "task=17",
        () -> "Страны",
        (String link) -> countriesHtml(text))
    .add() -> NumberUtils.isNumber(countryId),
        () -> "country_id=" + countryId,
        () -> Ob0.getOb(mains, countryId).getAt(1000000) + ": регионы",
        (String link) -> regionsHtml(text, countryId))
    .add() -> NumberUtils.isNumber(regionId),
        () -> "region_id=" + regionId,
        () -> Ob0.getOb(mains, regionId).getAt(1000098) + ": города",
        (String link) -> citiesHtml(text, link, regionId))
    .add() -> NumberUtils.isNumber(cityId),
        () -> "city_id=" + cityId,
        () -> Ob0.getOb(mains, cityId).getAt(1000098) + ": отели",
        (String link) -> hotelsHtml(text, link, cityId))
    .add() -> NumberUtils.isNumber(hotelId),
        () -> "hotel_id=" + hotelId,
        () -> "Отель \'" + Ob0.getOb(mains, hotelId).getAt(1990410000) + "\'",
        (String link) -> roomsHtml(text, link, hotelId))
    .add() -> NumberUtils.isNumber(roomId),
        () -> "room_id=" + roomId,
        () -> "Комп \'" + Ob0.getOb(mains, roomId).getAt(1000168) + "\'",
        (String link) -> nssHtml(text, link, roomId))
    .add() -> NumberUtils.isNumber(nsId),
        () -> "ns_id=" + nsId,
        () -> "НС \'" + Ob0.getOb(mains, nsId).getAt(1000348) + "\'",
        (String link) -> quotesHtmlV2(text, link, nsId))
    .add() -> NumberUtils.isNumber(quoteId),
        () -> "quote_id=" + quoteId,
        () -> "Квота ID " + quoteId,
        (String link) -> quoteEditHtmlV2(text, link, quoteId))
    .run(text);
printHtmlWithAuth(userEMail, "Авторизация - Пять", text.toString());
}

public void countriesHtml(StringBuilder text) throws Exception {
    Obb filter = Ob0.createFilter(4);
    Obb[] countries = Ob0.getSrcObs(mains, filter, 0, 0);
    Arrays.sort(countries, Comparator.comparing((Obb country) -> country.getAt(1000000)));
    text.append("<table class=\"data\"><tr><td>№</td><td>Название</td></tr>");
    int i = 0;
    for(Obb country : countries) {
        text.append(String.format("<tr><td>%d</td><td><a
href=\"?task=17&country_id=%s\">%s</a></td></tr>",
            ++i, country.id, country.getAt(1000000)));
    }
    text.append("</table>");
}

public void regionsHtml(StringBuilder text, String countryId) throws Exception {
    Map<String, String> regions = getRegions(countryId);
    List<String> regionIds = intern.Utils.getKeysSortedByValue(regions, false);
    text.append("<table class=\"data\"><tr><td>№</td><td>Название</td></tr>");
    int i = 0;
    for(String id : regionIds) {
        text.append(String.format("<tr><td>%d</td><td>\" +
            "<a href=\"?task=17&country_id=%s&region_id=%s\">%s</a></td></tr>",

```

```

        ++i, countryId, id, regions.get(id)));
    }
    text.append("</table>");
}

public void citiesHtml(StringBuilder text, String link, String regionId) throws Exception {
    Obb filter = Ob0.createFilter(5);
    Ob0.addCondition(filter, 1005101368, Ob0.ComparisonType.EQ, regionId);
    Obb[] cities = Ob0.getSrcObs(mains, filter, 0, 0);
    Arrays.sort(cities, Comparator.comparing((Obb city) -> city.getAt(1000098)));
    text.append("<table class=\"data\"><tr><td>№</td><td>Название</td></tr>");
    int i = 0;
    for(Obb city : cities) {
        text.append(String.format("<tr><td>%d</td><td><a href=\"%s&city_id=%s\">%s</a></td></tr>",
            ++i, link, city.id, city.getAt(1000098)));
    }
    text.append("</table>");
}

public void hotelsHtml(StringBuilder text, String link, String cityId) throws Exception {
    Obb filter = Ob0.createFilter(990);
    Ob0.addCondition(filter, 1990100059, Ob0.ComparisonType.EQ, cityId);
    Obb[] hotels = Ob0.getSrcObs(mains, filter, 0, 0);
    Arrays.sort(hotels, Comparator.comparing((Obb hotel) -> hotel.getAt(1990410000)));
    text.append("<table class=\"data\"><tr><td>№</td><td>Название</td><td>НСы</td></tr>");
    int i = 0;
    for(Obb hotel : hotels) {
        text.append(String.format("<tr><td>%d</td><td><a href=\"%s&hotel_id=%s\">%s</a></td><td>%s</td></tr>",
            ++i, link, hotel.id, hotel.getAt(1990410000),
            String.join(", ", hotel.getAts(1990423125))));
    }
    text.append("</table>");
}

public void roomsHtml(StringBuilder text, String link, String hotelId) throws Exception {
    Obb filter = Ob0.createFilter(21);
    Ob0.addCondition(filter, 1000169, Ob0.ComparisonType.EQ, hotelId);
    Obb[] rooms = Ob0.getSrcObs(mains, filter, 0, 0);
    Arrays.sort(rooms, Comparator.comparing((Obb hotel) -> hotel.getAt(1000168)));
    text.append("<table class=\"data\"><tr><td>№</td><td>Название</td></tr>");
    int i = 0;
    for(Obb room : rooms) {
        text.append(String.format("<tr><td>%d</td><td><a href=\"%s&room_id=%s\">%s</a></td></tr>",
            ++i, link, room.id, room.getAt(1000168)));
    }
    text.append("</table>");
}

public void nssHtml(StringBuilder text, String link, String roomId) throws Exception {
    Obb filter = Ob0.createFilter(46);
    Ob0.addCondition(filter, 1000350, Ob0.ComparisonType.EQ, roomId);
    Obb[] nss = Ob0.getSrcObs(mains, filter, 0, 0);
    Arrays.sort(nss, Comparator.comparing((Obb ns) -> ns.getAt(1000348)));
    text.append("<table class=\"data\"><tr><td>№</td><td>Название</td></tr>");
    int i = 0;
    for(Obb ns : nss) {
        text.append(String.format("<tr><td>%d</td><td><a href=\"%s&ns_id=%s\">%s</a></td></tr>",
            ++i, link, ns.id, ns.getAt(1000348)));
    }
    text.append("</table>");
}

```

```

}

public void quotesHtmlV2(StringBuilder text, String link, String nsId) throws Exception {
    Obb filter = Ob0.createFilter(990);
    Ob0.addCondition(filter, 1000117, Ob0.ComparisonType.EQ, nsId);
    Obb[] hotels = Ob0.getSrcObs(mains, filter, 0, 0);
    Arrays.sort(hotels, Comparator.comparing((Obb hotel) -> hotel.getAt(1000127)));
    text.append("<table class=\"data\"><tr><td>№</td><td>Название</td></tr>");
    int i = 0;
    for(Obb hotel : hotels) {
        text.append(String.format("<tr><td>%d</td><td><a href=\"%s&hotel_id=%s\">%s</a></td></tr>",
            ++i, link, hotel.id, hotel.getAt(1000127)));
    }
    text.append("</table>");
}

public void quoteEditHtmlV2(StringBuilder text, String link, String quoteId) throws Exception {
    Obb filter = Ob0.createFilter(990);
    Ob0.addCondition(filter, 1000117, Ob0.ComparisonType.EQ, quoteId);
    Obb[] hotels = Ob0.getSrcObs(mains, filter, 0, 0);
    Arrays.sort(hotels, Comparator.comparing((Obb hotel) -> hotel.getAt(1000127)));
    text.append("<table class=\"data\"><tr><td>№</td><td>Название</td></tr>");
    int i = 0;
    for(Obb hotel : hotels) {
        text.append(String.format("<tr><td>%d</td><td><a href=\"%s&hotel_id=%s\">%s</a></td></tr>",
            ++i, link, hotel.id, hotel.getAt(1000127)));
    }
    text.append("</table>");
}

public static class StackPager {
    private final List<Supplier<Boolean>> predicates = new ArrayList<>();
    private final List<Supplier<String>> links = new ArrayList<>();
    private final List<Supplier<String>> names = new ArrayList<>();
    private final List<Consumer<String>> consumers = new ArrayList<>();

    StackPager add(Supplier<Boolean> predicate, Supplier<String> link,
        Supplier<String> name, Consumer<String> consumer) {
        predicates.add(predicate);
        links.add(link);
        names.add(name);
        consumers.add(consumer);
        return this;
    }

    void run(StringBuilder text) throws Exception {
        StringBuilder link = new StringBuilder();
        StringBuilder name = new StringBuilder();
        int n = 0;
        while(++n < predicates.size() && predicates.get(n).get());
        for(int i = 0; i < n; ++i) {
            link.append(i == 0 ? "eldar?" : "&").append(links.get(i).get());
            name.append(i == 0 ? "" : " -> ").append(i < n - 1 ? String.format("<a href=\"%s\">", link) : "")
                .append(names.get(i).get()).append(i < n - 1 ? "</a>" : "");
        }
        text.append("<div style=\"text-align:center; margin-bottom:20px;\">").append(name).append("</div>");
        consumers.get(n - 1).accept(link.toString());
    }
}

public interface Supplier<T> {

```

```

    T get() throws Exception;
}

public interface Consumer<T> {
    void accept(T arg) throws Exception;
}

public void printHtmlWithAuth(String userEmail, String title, String text) throws Exception {
    String task = request.getParameter("task");
    authTask(userEmail != null);
    StringBuilder output = new StringBuilder();
    output.append("<div style=\"text-align:center;margin-bottom:20px;\">");
    if(userEmail == null) {
        output.append("<a href=\"?task=" + task + "&auth=need\">Авторизоваться</a>");
    } else {
        output.append("Привет, " + userEmail + "!");
    }
    output.append("</div>");
    printHtml(title, output + text);
}

public void authTask(boolean isAuth) throws Exception {
    String authParameter = request.getParameter("auth");
    boolean doYouNeedAuth = authParameter != null && authParameter.equals("need");
    boolean doYouNeedQuit = authParameter != null && authParameter.equals("quit");
    if(!isAuth && doYouNeedAuth) {
        AuthServlet.isAuth(request, response, mains, "meta");
    }
    if(isAuth && doYouNeedQuit) {
        String task = request.getParameter("task");
        response.setContentType("text/html; charset=UTF-8");
        Cookie[] cookies = request.getCookies();
        if(cookies != null) {
            for (var cookie : cookies) {
                cookie.setValue("");
                cookie.setPath("/");
                cookie.setMaxAge(0);
                response.addCookie(cookie);
            }
        }
        response.sendRedirect("eldar?task=" + task);
    }
}

public boolean hasRole(String role) {
    if(getUserEmail() == null) {
        return false;
    }
    Obb obUser = getUser();
    return Base.userHasRole(mains, Ob0.metaconnname, obUser, role);
}

public int hasTitle(String role) {
    if(getUserEmail() == null) {
        return -5;
    }
    User whoami = AuthServlet.isAuth(request, response, mains, "meta");
    Obb obUser = PersonalPageServlet.getUserByComm(mains, whoami);
    return Base.userHasRole(mains, Ob0.metaconnname, whoami, obUser, "100718258857",
        request.getHeader("X-Real-IP"), request.getHeader("User-Agent"));
}

```

```

public Obb getUser() {
    User whoami = AuthServlet.isAuth(request, response, mains, "meta");
    return PersonalPageServlet.getUserByComm(mains, whoami);
}

public String getUserEMail() {
    Cookie[] cookies = request.getCookies();
    boolean A100 = false, Z100 = false;
    String L = null;
    if(cookies != null) {
        for (var cookie : cookies) {
            if (cookie.getName().equals("A100") && !cookie.getValue().equals("")) {
                A100 = true;
            }
            if (cookie.getName().equals("Z100") && !cookie.getValue().equals("")) {
                Z100 = true;
            }
            if (cookie.getName().equals("L") && !cookie.getValue().equals("")) {
                L = cookie.getValue();
            }
        }
    }
    return A100 && Z100 ? L : null;
}

public void cheatCode() throws Exception {
    response.setContentType("application/json; charset=UTF-8");
    String type = request.getParameter("type");
    String s_begin = request.getParameter("begin");
    String s_count = request.getParameter("count");
    String s_id = request.getParameter("id");
    String s_atid = request.getParameter("at");
    String s_val = request.getParameter("val");
    int begin, count;
    Obb[] obbs;
    if(NumberUtils.isNumber(s_id)) {
        obbs = new Obb[]{Ob0.getOb(mains, s_id)};
    } else {
        if(!NumberUtils.isNumber(type)) {
            return;
        }
        begin = !NumberUtils.isNumber(s_begin) ? 0 : Integer.parseInt(s_begin);
        count = !NumberUtils.isNumber(s_count) ? 0 : Integer.parseInt(s_count);
        Obb filter = Ob0.createFilter(Integer.parseInt(type));
        if(NumberUtils.isNumber(s_atid) && NumberUtils.isNumber(s_val)) {
            Ob0.addCondition(filter, Integer.parseInt(s_atid), Ob0.ComparisonType.EQ, s_val);
        }
        obbs = Ob0.getSrcObs(mains, filter, begin, count);
    }
    out.print("[");
    int i = 0;
    for(Obb obb : obbs) {
        out.printf("{\"id\":\"%s\",\"id_user\":\"%s\",", obb.id, obb.id_user);
        int j = 0;
        for(String key : obb.zn.keySet()) {
            out.printf("\"%s\":\"%s\"%s", key, obb.getAt(key), ++j < obb.zn.size() ? "," : "");
        }
        out.printf("}%s", ++i < obbs.length ? "," : "");
    }
    out.print("]");
}

```



```

}

public void cheatFunction() throws Exception {
    /*Obb ob = Ob0.getOb(mains, "103610006184");
    ob.zn.remove("1799210047");
    ob.zn.remove("1799510027");
    ob.id_user = myId;
    Ob0.edtOb(mains, ob);
    out.print(ob.zn);*/
}

public void printHtml(String title, String text) {
    response.setContentType("text/html; charset=UTF-8");
    String task = request.getParameter("task");
    StringBuilder output = new StringBuilder();
    output.append(String.format("<!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; }" +
        "ul { list-style: none; }" +
        "ul li:before { content: \"—\"; position: relative; left: -5px; }" +
        "</style></head>" +
        "<body><div style=\"text-align:center;margin-bottom:20px;\">" +
        "<table class=\"form\">", title));
    int tasks = 0;
    for(String key : pagesMap.keySet()) {
        output.append(String.format("<tr>" +
            "<td style=\"font-weight:bold;color:#072d78;padding-right:15px;\">" +
            "%s</td><td>", key));
        List<String> numbers = pagesMap.get(key);
        for(int i = 1, n = numbers.size(); i <= n; ++i) {
            if(String.valueOf(tasks + i).equals(task)) {
                output.append(String.format("<b style=\"color:grey\">%s</b>%s",
                    numbers.get(i - 1), i < n ? " | " : ""));
            } else {
                output.append(String.format("<a href=\"?task=%d\">%s</a>%s",
                    tasks + i, numbers.get(i - 1), i < n ? " | " : ""));
            }
        }
        output.append("</td></tr>");
        tasks += numbers.size();
    }
    output.append(String.format("</table></div>%s</body></html>", text));
    out.print(output);
}
}

```