

**ВЕСТНИК**

**КЕМЕРОВСКОГО  
ГОСУДАРСТВЕННОГО  
УНИВЕРСИТЕТА**

**Выпуск 2 (10)**



**BULLETIN**

**OF KEMEROVO  
STATE UNIVERSITY**

**Issue 2 (10)**

**КЕМЕРОВО  
2002**



№ 2 (10)

Издается с 1999 г.

Журнал теоретических и  
прикладных исследований  
Выходит 1 раз в квартал

**СОДЕРЖАНИЕ**

<b>РАЗДЕЛ 1. БИОЛОГИЯ.....</b>	<b>6</b>
<i>Л.О. Петункина, Л.М. Лыбина, М.Д. Есмагамбетов</i> БЕРЕЗА БОРОДАВЧАТАЯ В ЗЕЛЕННЫХ НАСАЖДЕНИЯХ ГОРОДА КЕМЕРОВА.....	6
<i>А.М. Прохорова, Е.С. Гольдшмидт, Е.В. Броздецкая</i> ИНДИВИДУАЛЬНО- ТИПОЛОГИЧЕСКИЕ ОСОБЕННОСТИ ФУНКЦИОНАЛЬНОЙ АСИММЕТРИИ МОЗГА У СТУДЕНТОВ .....	12
<i>О.Л. Шиленок</i> ФАУНА СЛЕПНЕЙ (DIPTERA, TABANIDAE) ГОРНОЙ ШОРИИ.....	18
<i>Д.А. Золотарев</i> ИЗМЕНЕНИЕ СТРУКТУРЫ НАСЕЛЕНИЯ ПОЛУЖЕСТКОКРЫЛЫХ В ГОРОДСКИХ ЛАНДШАФТАХ.....	22
<i>С.В. Овсянникова, В.Б. Батурина</i> ОСОБЕННОСТИ ГУМИФИКАЦИИ ПОДЗЕМНЫХ ЧАСТЕЙ ЗЛАКОВЫХ И БОБОВЫХ ТРАВ И ИХ СМЕСЕЙ ПРИ БИОЛОГИЧЕСКОЙ РЕКУЛЬТИВАЦИИ НАРУШЕННЫХ ЗЕМЕЛЬ .....	27
<i>Л.О. Петункина</i> ВКЛАД ПРЕДПРИЯТИЙ ЭНЕРГЕТИКИ В ЗАГРЯЗНЕНИЕ ГОРОДА И ДЕЙСТВИЕ ПРИОРИТЕТНЫХ ЗАГРЯЗНИТЕЛЕЙ НА КОМПОНЕНТЫ ОКРУЖАЮЩЕЙ СРЕДЫ.....	32
<i>Н.И. Еремеева</i> ИЗМЕНЕНИЕ ПЛОТНОСТИ ПОПУЛЯЦИЙ ТОПОЛЕВОЙ МОЛИ- ПЕСТРЯНКИ В УСЛОВИЯХ ГОРОДА КЕМЕРОВО .....	37
<i>Н.И. Еремеева, С.В. Блинова</i> ВИДОВОЙ СОСТАВ И ОСОБЕННОСТИ ПОСЕЛЕНИЯ МУРАВЬЁВ В УРБАНИЗИРОВАННЫХ ЦЕНОЗАХ.....	43
<i>Д.В. Сущев</i> РЕДКИЕ ВИДЫ БУЛАВОУСЫХ ЧЕШУЕКРЫЛЫХ.....	48
КУЗНЕЦКО-САЛАИРСКОЙ ГОРНОЙ ОБЛАСТИ.....	48
<i>Л.Н. Ковригина</i> МОРФОЛОГИЧЕСКИЕ ПОКАЗАТЕЛИ УСТОЙЧИВОСТИ ЯЧМЕНЯ К СТЕБЛЕВОМУ ПОЛЕГАНИЮ .....	53
<i>О.М. Ольшевская, А.В. Заушинцева</i> МОРФОЛОГИЧЕСКИЕ ОСОБЕННОСТИ ГИБРИДОВ ЯЧМЕНЯ .....	60
<i>Н.А. Егорова, Г.А. Гореликова</i> ВОДНЫЕ МИКРОБЫ - ФИКСАТОРЫ НЕОРГАНИЧЕСКОГО ЙОДА.....	71
<i>М.Б. Лавряшина, М.В. Ульянова, Т.А. Толочко, В.В. Карпова</i> ГЕНЕТИКО- ДЕМОГРАФИЧЕСКИЕ АСПЕКТЫ ФОРМИРОВАНИЯ ГОРОДОВ КУЗБАССА.....	77

<b>РАЗДЕЛ 2. ЮРИСПРУДЕНЦИЯ</b> .....	83
<i>Ю.В.Самович</i> ПРАВО НА СПРАВЕДЛИВОЕ СУДЕБНОЕ РАЗБИРАТЕЛЬСТВО ..... 83 (В КОНТЕКСТЕ ДЕЯТЕЛЬНОСТИ ЕВРОПЕЙСКОГО СУДА ПО ПРАВАМ ЧЕЛОВЕКА) .....	83
<i>Е.В.Жариков</i> ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ЛИЦ, СОДЕЙСТВУЮЩИХ ПРАВОСУДИЮ .....	88
<i>А.Ю.Пастушенко</i> ПРОБЛЕМЫ ПРЕДОСТАВЛЕНИЯ РЕЗУЛЬТАТОВ ОПЕРАТИВНО-РОЗЫСКНОЙ ДЕЯТЕЛЬНОСТИ НА РАЗЛИЧНЫХ СТАДИЯХ УГОЛОВНОГО ПРОЦЕССА .....	92
<i>В.К.Зникин</i> ОПЕРАТИВНО-РОЗЫСКНАЯ ДЕЯТЕЛЬНОСТЬ: ФОРМИРОВАНИЕ ПОНЯТИЯ И СОВРЕМЕННАЯ КЛАССИФИКАЦИЯ .....	99
<i>М.В.Безуглов, С.И.Неганов</i> ЗАКОНОМЕРНОСТИ ГОРЕНИЯ, УЧИТЫВАЕМЫЕ ПРИ РАССЛЕДОВАНИИ И СУДЕБНОМ РАССМОТРЕНИИ УГОЛОВНЫХ ДЕЛ О КРИМИНАЛЬНЫХ ПОЖАРАХ .....	103
<i>Н.А.Юркевич</i> СУЩНОСТЬ, ЗНАЧЕНИЕ И ЗАДАЧИ СТАДИИ ПРЕДАНИЯ СУДУ ПО ЗАКОНОДАТЕЛЬСТВУ РОССИЙСКОЙ ИМПЕРИИ.....	109
<i>С.О.Гаврилов</i> ВЫБОРЫ В МЕСТНЫЕ СОВЕТЫ 1939 г. ....	112
<i>Е.Ю.Мотова</i> ХОЗЯЙСТВУЮЩИЕ СУБЪЕКТЫ: КЛАССИФИКАЦИЯ ПОДХОДОВ .....	113
<i>Ю.А.Левич</i> НОВЫЙ УГОЛОВНО-ПРОЦЕССУАЛЬНЫЙ КОДЕКС РОССИЙСКОЙ ФЕДЕРАЦИИ – НОВАЯ ПОЛИТИКА ГОСУДАРСТВА В ОБЛАСТИ СУДОПРОИЗВОДСТВА .....	117
<i>Ю.А.Левич</i> ЗАКЛЮЧЕНИЕ ПОД СТРАЖУ ОБВИНЯЕМОГО СУДЬИ.....	126
<b>РАЗДЕЛ 3. ЭКОНОМИКА И МАТЕМАТИКА</b> .....	132
<i>Э.А.Аксенова, Н.И.Новиков</i> КОНКУРЕНТНЫЕ ВОЗМОЖНОСТИ МАШИНОСТРОИТЕЛЬНЫХ ПРЕДПРИЯТИЙ КУЗБАССА .....	132
<i>Н.И.Новиков</i> К ВОПРОСУ О КОНКУРЕНТОСПОСОБНОСТИ МЕТАЛЛОПРОДУКЦИИ ПРЕДПРИЯТИЙ ЧЕРНОЙ МЕТАЛЛУРГИИ ЗАПАДНО- СИБИРСКОГО РЕГИОНА .....	138
<i>В.К.Буторин</i> ЭКОНОМИКО-МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЗАДАЧ УПРАВЛЕНИЯ РЕСУРСАМИ ДЛЯ ПРЕДПРИЯТИЙ, НАХОДЯЩИХСЯ В КРИЗИСНЫХ УСЛОВИЯХ .....	145
<i>В.Г.Чуркин</i> КОНЦЕПЦИЯ СТРУКТУРЫ ИНФОРМАЦИОННОЙ БАЗЫ ДАННЫХ ОРГАНИЗАЦИИ .....	149
<i>В.О.Каледин, Т.В.Бурнышева</i> АНАЛИЗ ВЛИЯНИЯ КОЭФФИЦИЕНТА ПУАССОНА НА НАПРЯЖЕННО-ДЕФОРМИРОВАННОЕ СОСТОЯНИЕ МАССИВА ГОРНЫХ ПОРОД С ИСКРИВЛЕННЫМИ ПЛАСТАМИ.....	155
<i>В.А.Перминов</i> МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЗАГРЯЗНЕНИЯ ОКРУЖАЮЩЕЙ СРЕДЫ ОТ АВТОТРАНСПОРТА .....	159
<i>К.Ю.Войтиков, О.А.Змеев, А.Н.Моисеев</i> К ВОПРОСУ О РЕАЛИЗАЦИИ КЛАССИФИКАТОРА В ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЕ.....	167
<b>РАЗДЕЛ 4. ОБРАЗОВАНИЕ</b> .....	178
<i>В.А.Саркисян</i> РОЛЬ ГОСУДАРСТВЕННЫХ ОБРАЗОВАТЕЛЬНЫХ СТАНДАРТОВ В ФОРМИРОВАНИИ ПРОФЕССИОНАЛЬНОЙ ГОТОВНОСТИ СТУДЕНТОВ- ЗАОЧНИКОВ И ПОВЫШЕНИИ КАЧЕСТВА ИХ ПОДГОТОВКИ .....	178
<i>Е.Е.Адакин, В.А.Саркисян</i> ОПЫТ И ПЕРСПЕКТИВЫ РАЗВИТИЯ БЕЛОВСКОГО	

ИНСТИТУТА (ФИЛИАЛА) В РЕГИОНАЛЬНОЙ СИСТЕМЕ НЕПРЕРЫВНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ .....	183
<i>Е.Е.Адакин</i> КОНЦЕПТУАЛЬНЫЕ ОСНОВЫ ОРГАНИЗАЦИИ ПОДГОТОВКИ СПЕЦИАЛИСТОВ С ВЫСШИМ ПРОФЕССИОНАЛЬНЫМ ОБРАЗОВАНИЕМ В РАМКАХ ИНСТИТУТА-ФИЛИАЛА .....	192
<i>Е.Е.Адакин, К.А.Кабанов</i> КРАЕВЕДЕНИЕ В ЕДИНОЙ ТРУДОВОЙ ШКОЛЕ В ПЕРВЫЕ ГОДЫ НЭПА .....	197
<i>В.П.Клочков, Л.В.Ткаченко, Л.П.Михальцова</i> ПРИНЦИП И КРИТЕРИИ ОПТИМИЗАЦИИ УПРАВЛЕНИЯ ОБЩЕОБРАЗОВАТЕЛЬНОЙ СФЕРОЙ .....	203
<i>Е.В.Вечер</i> СИБИРСКАЯ ЖЕЛЕЗНОДОРОЖНАЯ МАГИСТРАЛЬ КАК ОДИН ИЗ ФАКТОРОВ РАЗВИТИЯ ОБРАЗОВАНИЯ И ПРОСВЕЩЕНИЯ В ЗАПАДНОЙ СИБИРИ НА РУБЕЖЕ XIX–XX ВВ. ....	211
<i>Т.С.Панина, Н.Г.Шульга</i> ПОИСКОВЫЕ ХАРАКТЕРИСТИКИ РЕГИОНАЛЬНОГО УПРАВЛЕНИЯ ОБЩЕОБРАЗОВАТЕЛЬНОЙ СФЕРОЙ.....	219
<i>В.П.Клочков, Л.Б.Проскуракова, У.П.Яковенко</i> СИСТЕМОПОРОЖДАЮЩИЕ И СИСТЕМОУСЛОВЛИВАЮЩИЕ ФАКТОРЫ ОПТИМИЗАЦИИ СОВМЕСТИМОСТИ УЧЕБНЫХ ТЕКСТОВ.....	230
<i>Л. Б.Проскуракова</i> СИСТЕМНЫЙ ПОДХОД К АНАЛИЗУ СОВМЕСТИМОСТИ УЧЕБНЫХ ТЕКСТОВ .....	237

8. Penner I.E., Haselman L.C., Edwards L.L. Buoyant plume calculation // AIAA Paper.- 1985.- М 459. - 9р.
9. Патанкар С. Численные методы решения задач теплообмена и динамики жидкости. - М.: Энергоатомиздат. 1984.
10. Перминов В.А., Шипулина О.В. О численном решении некоторых задач теории лесных пожаров // Физическая газодинамика в реагирующих средах. - Новосибирск: Наука, 1990.-С. 158-169.
11. Stone H.L. Iterative solution of implicit approximations of multi-dimensional partial differential equations // SIAM J. Num. Anal., 5.-P.530-558.
12. База данных по уровням нагрузок автотранспортных систем на окружающую среду. - М.: ВНИИПРИРОДА, 1991.

**Войтиков К.Ю., Змеев О.А., Моисеев А.Н.  
К ВОПРОСУ О РЕАЛИЗАЦИИ КЛАССИФИКАТОРА В ОБЪЕКТНО-  
ОРИЕНТИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЕ**

Современные тенденции объектно-ориентированного проектирования сложных информационных систем предлагают использовать для их реализации трехуровневую архитектуру. В настоящей работе используется подход к реализации многоуровневой архитектуры распределенного приложения, предложенный в [1]. Основная идея данного подхода заключается в логическом и физическом распределении обязанностей объектов системы по трем основным видам пакетов (интерфейс, предметная область, база данных), при этом на каждом уровне реализуется типичная для него функциональность соответствующего объекта. Так интерфейсная часть реализует визуальное отображение, заполнение данными и инициализацию команд пользователя системы, связанных с объектом. Объекты предметной области реализуют бизнес-логику приложения, обеспечивают взаимосвязи и диспетчеризацию потоков команд между другими пакетами. Уровень хранения отвечает за инициализацию объекта из базы данных, сохранение в ней новых объектов и внесенных изменений.

Таким образом, любая сущность предметной области в такой системе реализуется в виде совокупности трех классов, каждый из которых выполняет свою часть функциональных обязанностей и располагается в соответствующем пакете приложения. Рассмотрим реализацию этого подхода на примере реализации подсистемы нормативно-справочной информации, без которой не обходится ни одна современная информационная система. При этом в рамках настоящей работы будут рассмотрены только два

уровня архитектуры: уровень интерфейса и уровень предметной области. Организация уровня хранения является отдельной проблемой и в рамках настоящей работы не рассматривается.

В литературе встречается достаточно много определений для классов, обеспечивающих функциональность этой подсистемы, например, справочник, перечисление и т.д. В теории баз данных соответствующие структуры получили название классификаторов, в настоящей работе будет использоваться именно этот термин. Под классификатором в дальнейшем будет пониматься объект информационной системы, содержащий набор (список) значений условно-постоянных реквизитов для каких-либо других объектов приложения. Отдельный элемент такого списка в дальнейшем будет называться элементом классификатора.

Согласно приведенному выше определению, функциональность классификатора может быть распределена по двум основным блокам. Первый блок включает в себя отображение списка значений в удобном, согласно концепции интерфейса системы, виде. Это может быть окно, диалог, выпадающий список и т.д. Естественно реализовать соответствующую функциональность в классе, представляющем классификатор на уровне интерфейса. Задача второго блока - организация управления данными: внесение изменений, удаление, загрузка и сохранение в базе данных, поиск и т.д. реализуется классами-классификаторами на уровне хранения и предметной области. При этом должна обеспечиваться функциональная целостность объекта между соответствующими представлениями на каждом уровне информационной системы.

Определим интерфейс класса, реализующего классификатор, на уровне взаимодействия с пользователем. В общем случае список операций этого интерфейса должен поддерживать:

- 1) запрос данных классификатора (списка целиком) у сервера;
- 2) отображение данных;
- 3) внесение нового элемента;
- 4) изменение данных (редактирование) элемента;
- 5) удаление элемента;
- 6) сохранение изменений в базе данных;
- 7) поиск элемента;
- 8) сортировка списка;
- 9) фильтрация данных по некоторому признаку;
- 10) подстановка значения (ссылки) выбранного элемента в реквизит объекта, инициировавшего эту операцию.

Сразу оговоримся, что в дальнейшем не будем описывать реализацию следующих операций: 2, 7, 8 и 9 – так как практически все современные средства разработки предлагают стандартные варианты их реализации, 5 – так как реализация этой операции неразрывно связана с организацией ссылочной целостности данных и требует отдельного рассмотрения; 6 – предполагая, что все изменения сохраняются непосредственно после выполнения соответствующей операции 3-5.

Вернемся к некоторым аспектам архитектуры описываемого приложения. В [1] предлагается использовать в качестве посредников между уровнями приложения классы, оформленные согласно шаблону «Контроллер» [2]. Таким образом, далее предполагаем, что на каждом уровне реализован соответствующий класс, обеспечивающий доступ к объектам своего уровня и связь с контроллерами других уровней. Еще один важный момент – все объекты системы, включая классификаторы и их элементы, имеют уникальные идентификаторы.

Итак, опишем реализацию на уровне интерфейса с пользователем, реализацию операций выборки данных классификатора, внесения нового элемента, редактирования элемента, подстановки значения в другой объект.

Операция 1 «Выборка данных» может быть описана следующей диаграммой (рис. 1). По запросу на инициализацию классификатора объект Контроллер клиента производит поиск в коллекции открытых на уровне клиента объектов по указанному в запросе идентификатору. В случае если классификатор найден, Контроллер клиента возвращает ссылку на него, иначе Контроллер клиента обращается к Контроллеру сервера предметной области с запросом и на основе полученных от сервера данных создает объект Классификатор, размещает его в своем списке объектов и возвращает в приложение ссылку на него.

Операции 3 и 4 предполагают, что объект Классификатор уже существует. При этом «Внесение нового элемента» (рис. 2) предполагает создание окна редактирования, заполнение его соответствующими данными (значениями «по умолчанию»), показ окна и в случае подтверждения ввода – передачу элемента серверу. Так как при этом элементу на уровне интерфейсного приложения не может быть присвоено глобального уникального идентификатора Системы, то поле идентификатора остается пустым и заполняется сервером. Таким образом, эти сведения Контроллер принимает от сервера предметной области и вносит в соответствующий элемент справочника.

Операция «Редактирование элемента» (рис. 3) реализуется аналогичным образом, однако для заполнения элементов окна редактирования используются данные текущего элемента справочника. Кроме того, элемент уже имеет уникальный идентифи-

катор, а значит, новых данных от сервера принимать не нужно. Еще одно отличие связано с тем, что элемент, который берется на редактирование, может уже редактироваться другим пользователем Системы. Поэтому перед выполнением всей операции Контроллер обращается за разрешением на редактирование с указанием идентификатора элемента. Эта операция заодно используется сервером предметной области для блокировки записи (см. ниже).

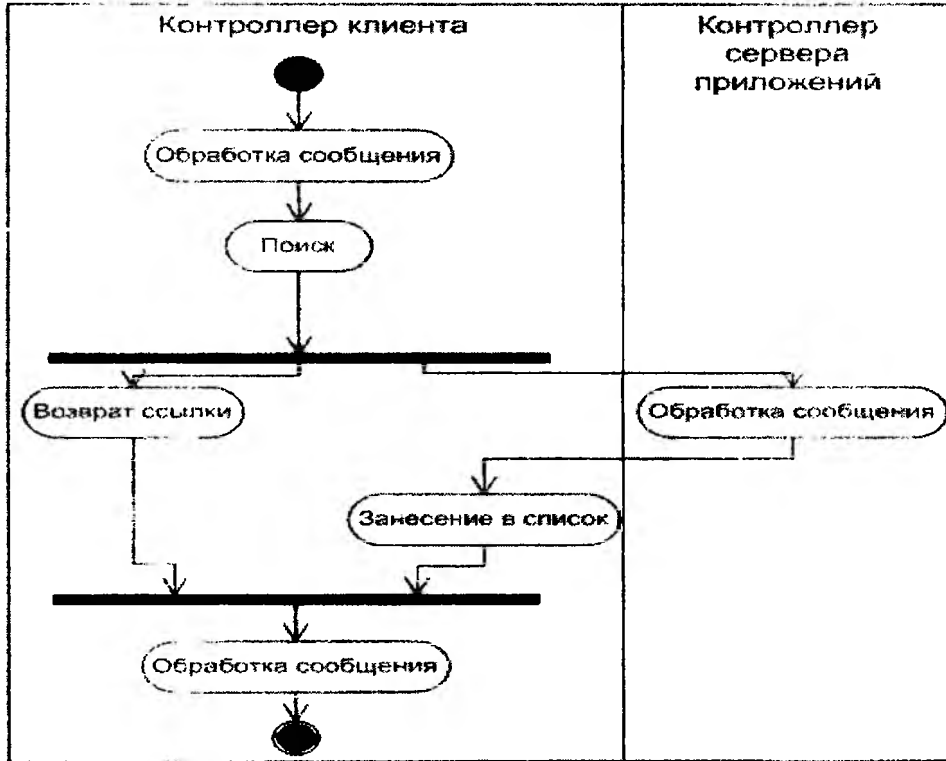


Рис. 1

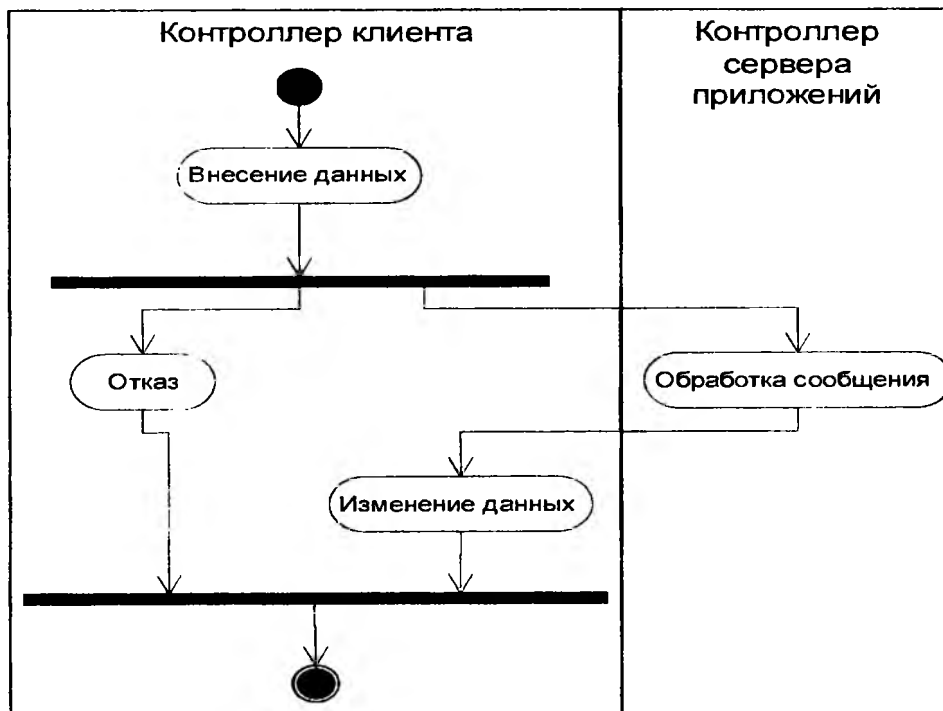


Рис. 2



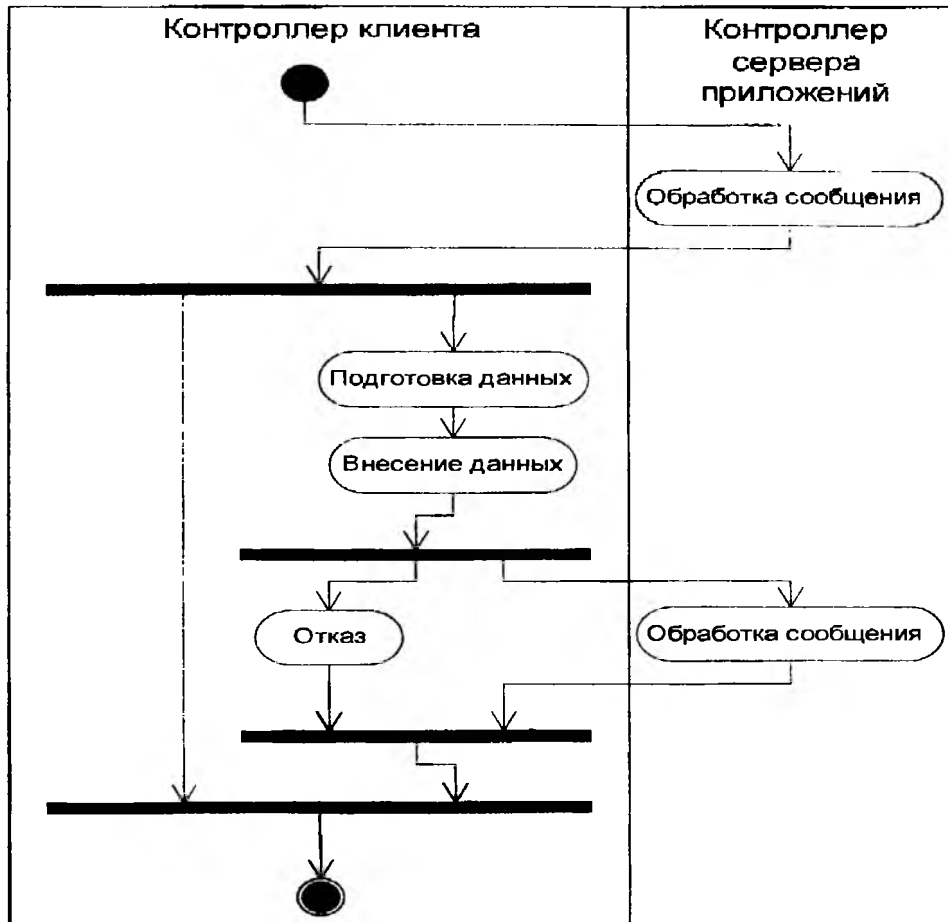


Рис. 3

Реализация операции 10 «Подстановка значения» может отличаться в зависимости от архитектуры приложения, но всегда предполагает выборку значения из справочника другим объектом. Поэтому последовательность действий Системы можно представить следующим образом (рис 4). Объект, требующий подстановки, обращается к Контроллеру с требованием показать окно выбора из справочника, указывая идентификатор Классификатора. При этом объект может указать и идентификатор элемента справочника, которому соответствует текущее значение его реквизита для позиционирования курсора в окне выбора. Контроллер производит операцию 1 «Выборка данных», сообщает объекту Классификатор о позиционировании курсора и заставляет его модально показать окно выбора. В случае подтверждения выбора в окне Контроллер возвращает вызвавшему объекту новое значение идентификатора элемента справочника (возможно, и другие необходимые атрибуты этого элемента). В случае отмены выбора – предыдущее значение идентификатора либо информацию об отказе от выбора.

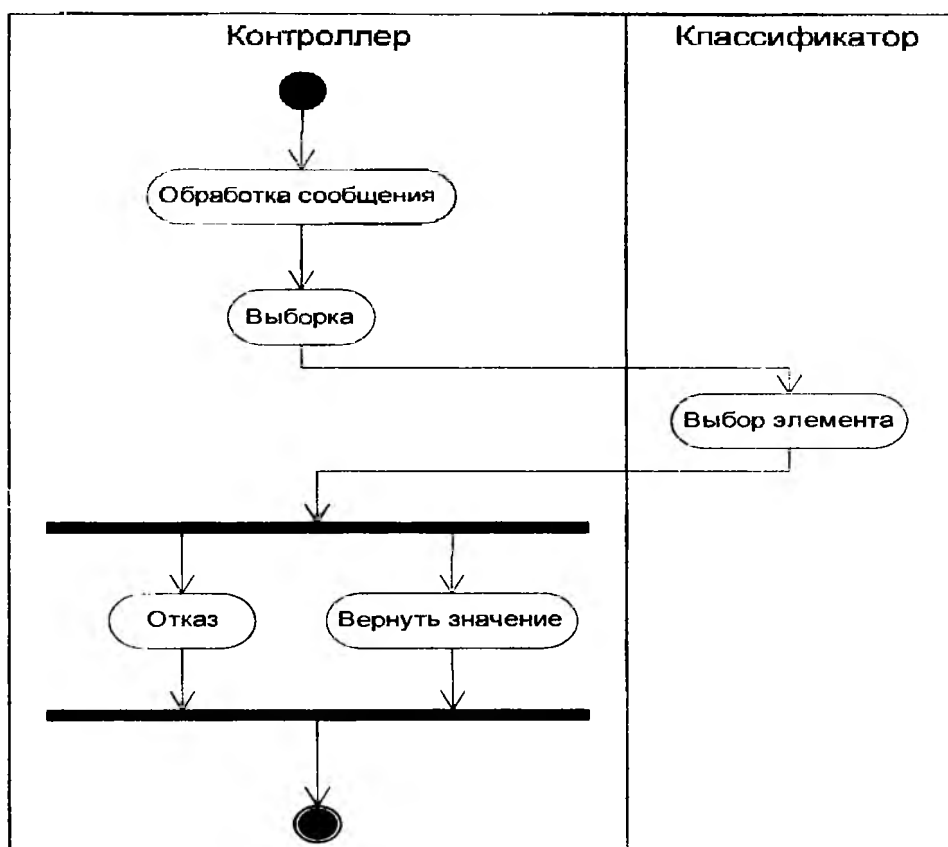


Рис. 4.

Теперь рассмотрим обязанности объекта, соответствующего классификатору на уровне сервера предметной области. Классификаторы обычно не предполагают реализацию какой-либо функциональности, касающейся именно предметной области. По этой и другим причинам ниже сервер предметной области [1] будем называть сервером приложений.

Таким образом, так же как для уровня интерфейса, определим необходимые операции, производимые со справочником на сервере приложений:

- 1) выборка данных;
- 2) определение прав доступа;
- 3) блокировки редактируемых элементов;
- 4) представление измененных данных.

Для реализации объекта Классификатор в приложении сервера введем следующие объекты:

- Пользователь – пользователь системы. Имеет следующие атрибуты: идентификатор, имя, права доступа.
- Сеанс – уникальный сеанс, выделенный системой пользователю для работы. Имеет следующие атрибуты: идентификатор, ссылка на пользователя. Этот объект нужен

для того, чтобы обеспечить работу одного и того же пользователя с разных рабочих мест.

- Подключение – объект, отвечающий за подключение к какому-либо классификатору. Атрибуты: идентификатор, ссылка на классификатор, ссылка на сеанс, способ использования. Совокупность этих объектов обеспечивает систему блокировок редактируемых элементов, а также обновление данных классификаторов в приложениях интерфейса.
- Классификатор – объект, отвечающий за представление классификатора на сервере приложений. Имеет следующие атрибуты: идентификатор, содержимое классификатора, список подключений.
- Контроллер – объект, через который осуществляется взаимодействие с клиентским приложением и сервером баз данных. Имеет следующие атрибуты: список пользователей, подключившихся к системе, список открытых классификаторов, список открытых документов.

Структурная диаграмма классов для сервера приложений показана на рисунке 5.

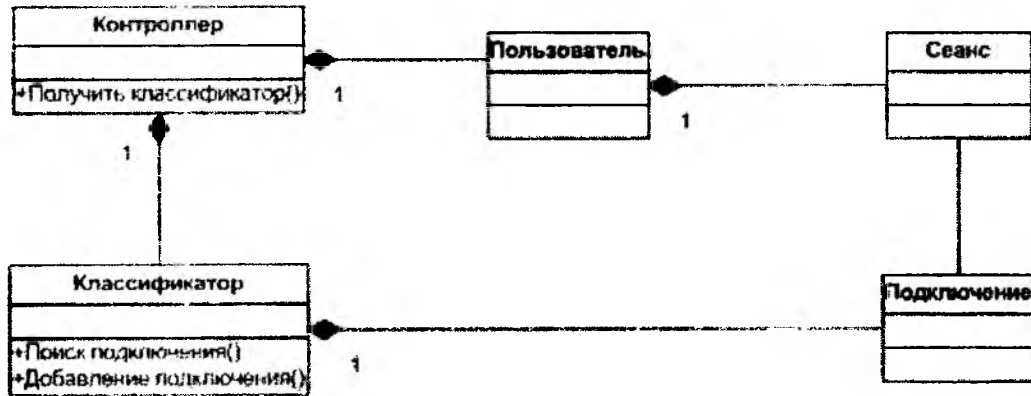


Рис. 5

Как уже было сказано выше, общение клиента с сервером приложений и сервером баз данных осуществляется при помощи объекта **Контроллер**, отвечающего требованиям шаблона [2]:

Прежде, чем начнем рассматривать реализацию классификатора, обсудим вопрос о правах доступа пользователей. При поступлении любой команды от клиента (открыть классификатор, внести изменения и т.д.) происходит проверка прав доступа данного пользователя. Если ему запрещена вызываемая операция, то клиенту возвращается код ошибки. В дальнейшем мы эту операцию будем опускать, но учитывать, что она выполняется каждый раз при обращении клиентского приложения к серверу. Диа-

грамма деятельности показана на рисунке 6.

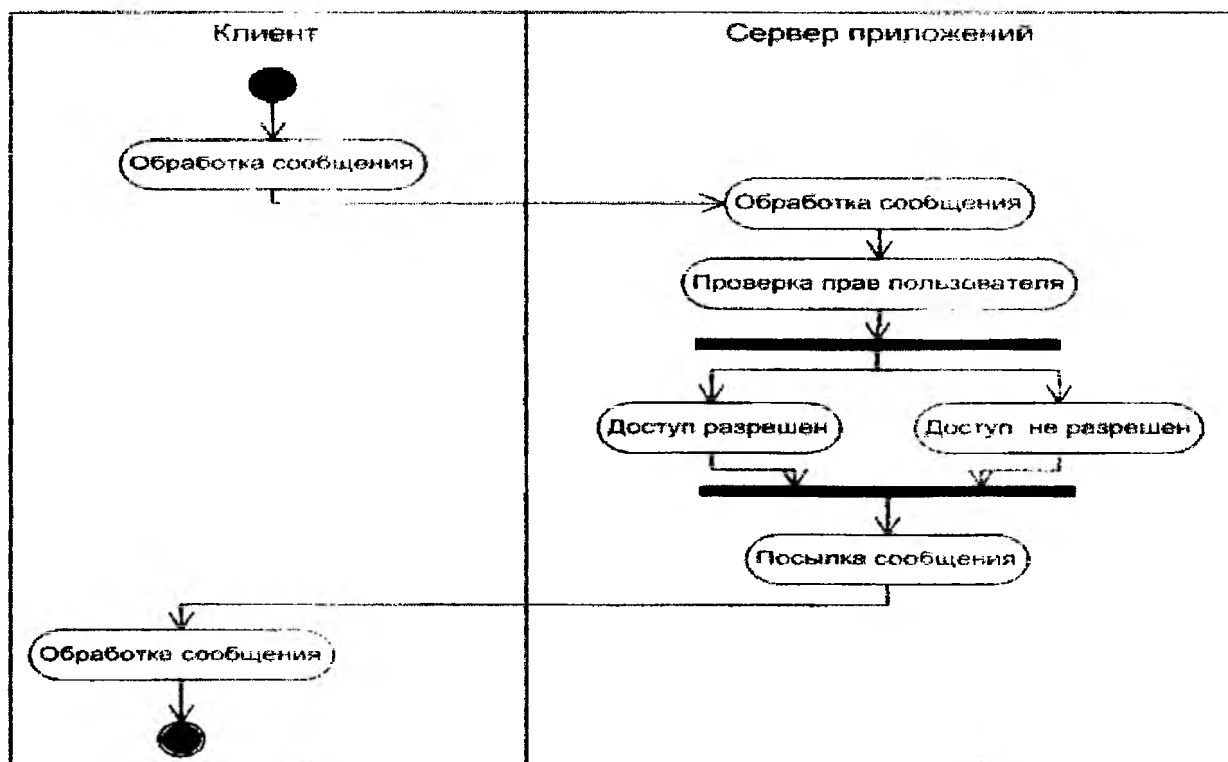


Рис. 6

Выборка классификатора осуществляется следующим образом. Клиент вызывает функцию Контроллера уровня интерфейса со следующими параметрами: команда об открытии классификатора, номер пользователя, номер классификатора. Полученная информация обрабатывается Контроллером сервера приложений. После расшифровки информации Контроллер ищет указанный классификатор в своем списке открытых Классификаторов. Поиск осуществляется по уникальному идентификатору. В результате поиска возможны две ситуации: либо Контроллер находит объект в своем списке открытых Классификаторов, либо не находит. Диаграмма деятельности для объекта Классификатор представлена на рисунке 7.

Рассмотрим первую ситуацию - указанный идентификатор найден в списке. Тогда в список подключений найденного объекта добавляется новое с указанием соответствующего идентификатора сеанса пользователя. После этого Контроллер сервера приложений отправляет клиенту в качестве ответа содержимое найденного Классификатора. Данный поиск осуществляется без обращения к базе данных, и, таким образом, вся обработка производится в оперативной памяти сервера, что существенно ускоряет выполнение операций.

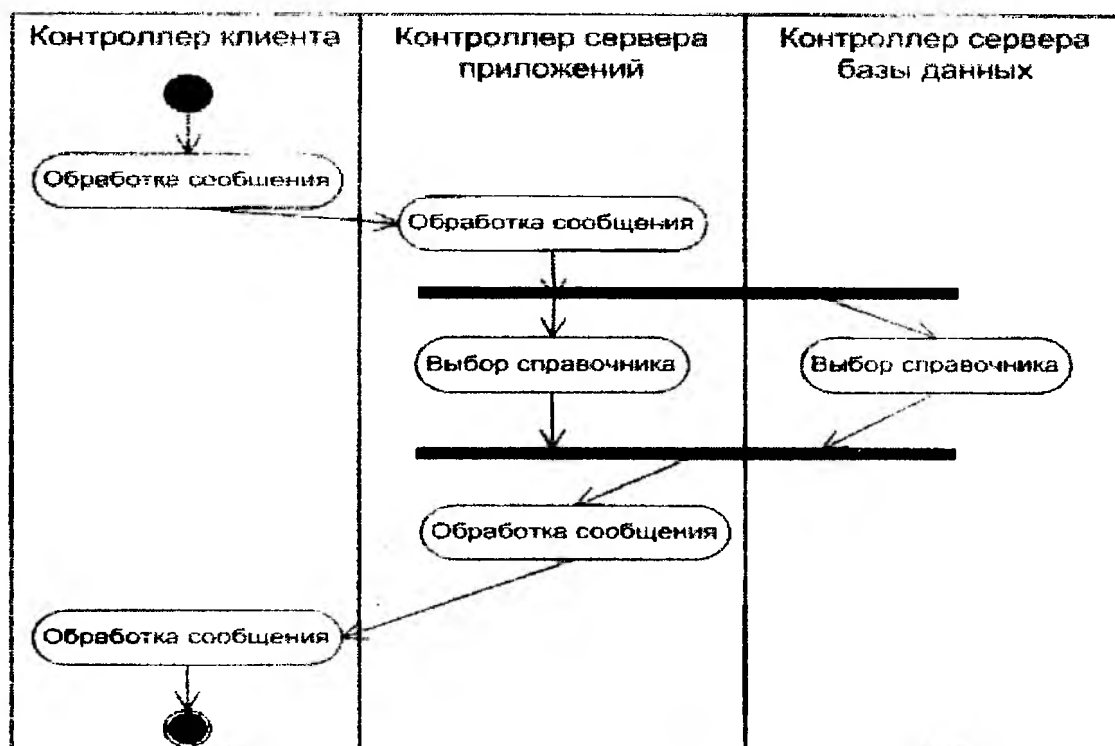


Рис. 7

Теперь рассмотрим ситуацию, когда поиск завершился неудачно, т.е. указанный Классификатор не найден в соответствующем списке Контроллера. Это говорит о том, что данный Классификатор открывается впервые. Тогда Контроллер сервера приложений переадресует команду Контроллеру сервера баз данных, который выберет необходимую информацию. После получения данных Контроллер создаст соответствующий экземпляр объекта Классификатор и внесет его в свой список открытых Классификаторов. Затем происходит добавление к Классификатору подключения сеанса пользователя. В завершение данные Классификатора передаются вызвавшему приложению.

Рассмотрим, что происходит при добавлении элемента в Классификатор или изменении существующего. В этом случае Контроллер сервера приложений переадресует команду контроллеру сервера баз данных. После сохранения информации, измененные данные приходят на сервер приложений, где Контроллер определяет, что же произошло: вставка новой записи либо изменение существующей. При вставке новой записи Контроллер в своем списке ищет нужный объект Классификатора и переадресует команду ему, после чего полученная информация добавляется в содержимое Классификатора. При изменении элемента Классификатор находит этот элемент в содержимом и заменяет его данные. После обработки информации сервером приложений Контроллер передает клиенту информацию об успешном завершении операции либо код ошибки в противном случае (а при добавлении элемента – присвоенный ему идентификатор).

После завершения операций, связанных с изменением данных, сервер приложений должен оповестить всех клиентов системы о произошедших изменениях в Классификаторе. Для этого после завершения на сервере приложений операций, связанных с изменением данных, в списке подключившихся к нему сеансов пользователей производится поиск сеансов, для которых изменения еще не отражены, и пересылка этим сеансам изменившихся данных.

При редактировании элементов Классификатора необходимо предотвратить возможность их редактирования другими пользователями. Для этого сервер приложений использует блокировки. Блокировка – это механизм, используемый для одновременного доступа нескольких транзакций различных пользователей к одной и той же базе данных. Для реляционных баз данных существуют несколько уровней блокировки. Блокировка на уровне таблицы запрещает модифицировать таблицы, участвующие в выполняющихся в настоящее время транзакциях. Блокировка на уровне страниц: сервер блокирует некоторые блоки на диске – так называемые страницы. Но эти методы не совсем удобны для пользователя, т.к. первый метод хотя и позволяет параллельную обработку, но достаточно медленный, потому что пользователям часто приходится обращаться к одним и тем же таблицам.

Некоторые серверы предлагают блокировку на уровне записи, т.е. отдельной строки таблицы базы данных. Считается, что данный метод связан с большими затратами ресурсов для хранения информации о блокировках. Ниже предлагается более эффективный метод решения данной проблемы.

Реализация блокировки на уровне отдельной записи осуществляется следующим образом. Клиент берет элемент Классификатора на редактирование. Одновременно с этим клиент посылает запрос Контроллеру сервера приложений о том, не заблокирована ли данная запись другим пользователем. Контроллер сервера приложений, получив такую команду, ищет в своем списке Классификаторов Классификатор с данным номером, а затем Классификатор ищет в своем списке подключений сеанс пользователя, который заблокировал эту запись. Если поиск завершился неудачей, то в список вносится пользователь, пославший команду блокировки, с пометкой, что он взял данную запись для редактирования. Теперь в случае, если другой пользователь попытается отредактировать или удалить данную запись, то сервер выдаст сообщение о том, что запись заблокирована другим пользователем. Соответствующая диаграмма для блокировки приведена на рисунке 8.

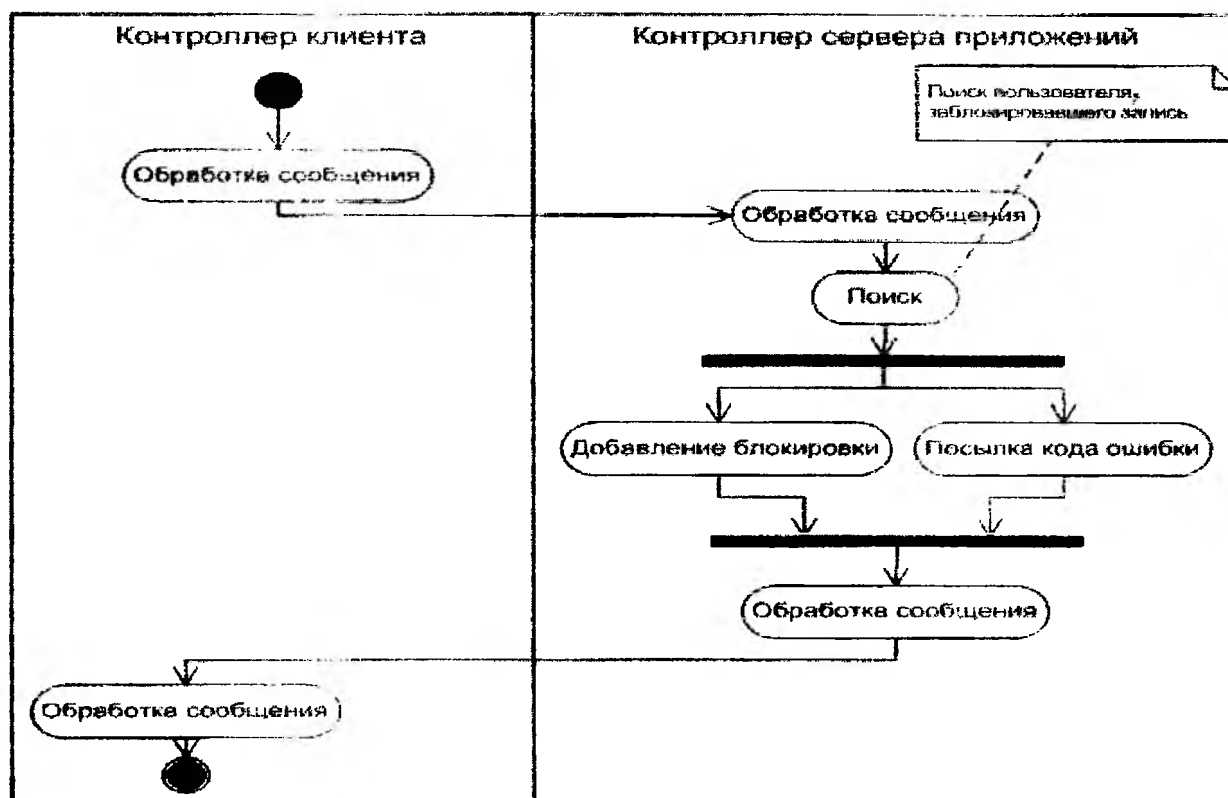


Рис. 8

Как уже отмечалось ранее, все данные о блокировках хранятся в оперативной памяти компьютера, что сводит затраты ресурсов к минимуму.

В данной работе подробно рассмотрены некоторые вопросы реализации классификаторов в объектно-ориентированной информационной системе, организованной на основе трехзвенной архитектуры. Поскольку при рассмотрении данного вопроса не конкретизируется тип классификатора, платформа, интерфейс и предметная область приложения, то можно сделать вывод о том, что предложенная схема может служить основой для разработки соответствующего шаблона и широко использоваться при проектировании и реализации распределенных информационных систем на основе объектно-ориентированного подхода.

#### Литература

1. Змеев О.А. Моисеев А.Н. Шаблон диаграммы компонентов информационной системы корпоративного уровня// Вестник ТГУ. 2002.
2. Ларман К. Применение UML и шаблонов проектирования.- М., 2001. – 496с.