

УО «ВГУ им. П.М. Машерова»
Кафедра прикладной математики и механики

С. А. Ермоченко

**Применение Java-технологии
для разработки информационных систем**

**Задания на лабораторные работы по дисциплине
«Технологии разработки информационных систем»
для студентов 4 курса специальности Прикладная математика
(1-31 03 03)**

Витебск, 2011

Лабораторная №1
Проектирование информационной системы
6 часов

1. Составить предварительный список функциональных требований по проекту. Согласовать его с преподавателем.
2. Провести предварительный анализ требований, уточнив их.
3. Разработать экраны пользователей для всех выделенных ролей (с предварительным дизайном HTML-страниц). Экраны пользователя должны отражать переходы с одной странице на другой при активизации определенного элемента управления (кнопки, ссылки и т.д.). Согласовать экраны пользователей с преподавателем. Пример см. в папке «**Shop/doc/user boards**».
4. Окончательно уточнить на основе разработанных экранов пользователей функциональные и другие требования к информационной системе. Согласовать требования с преподавателем. Пример см. в файле «**Shop/doc/requirements.html**».
5. На основе функциональных требований и экранов пользователей разработать классы–информационные эксперты, отражающие специфику предметной области. Согласовать спроектированные классы с преподавателем. Пример см. в пакете **entity**.
6. На основе разработанных классов спроектировать структуру базы данных, согласовать ее с преподавателем. Пример см. в папке: «**Shop/sql**».
7. Составить SQL-скрипты для наполнения тестовыми данными базы данных информационной системы. Пример см. в файле: «**Shop/sql/5_fill_tables.sql**»

Лабораторная №2
Data Access Object (DAO)
4 часа

1. Описать параметризованный интерфейс для DAO, определяющий четыре основные операции над данными в хранилище (БД).
2. Описать для каждого класса – информационного эксперта интерфейс, расширяющий базовый интерфейс для конкретной сущности (при необходимости, если точно известно из функциональных требований, в расширяющем интерфейсе описать дополнительные операции над данными).
3. С использованием шаблонов Singleton и Object Pool реализовать пул соединений к базе данных (для корректной загрузки драйвера MySQL с помощью вызова `Class.forName("com.mysql.jdbc.Driver")` необходимо при компиляции приложения указывать путь к файлу с библиотекой, содержащей драйвер `mysql-connector-java-5.1.17-bin.jar`. В IDE Eclipse этот путь можно задать в настройках проекта `Project > Properties` в дереве категорий в пункте `Java Build Path` на вкладке `Libraries` с помощью кнопки `Add JARs...`).
4. Реализовать все описанные DAO-интерфейсы для СУБД MySQL.
5. Разработать согласно шаблону Abstract Factory классы, создающие нужные реализации DAO-интерфейсов.
6. Написать несколько классов, тестирующих реализованные классы с использованием консоли.

Указания:

Работа с сервером MySQL осуществляется через утилиты в каталоге **bin**.

Для запуска сервера используется команда (без параметров):

mysqld.exe

Для остановки сервера используется команда:

mysqladmin.exe --user=root --password=root shutdown

Для подключения к серверу через консольный клиент используется команда:

mysql --user=<имя пользователя> --password=<пароль>

Далее можно выполнять обычные запросы на языке SQL.

Для более простого запуска и остановки сервера можно использовать инструмент Ant среды Eclipse. Для этого необходимо:

1. через главное меню открыть отдельную вкладку Ant (`Window > Show view > Ant`)
2. создать в корне проекта файл `build.xml` (пример файла см. **Shop/build.xml**). В данном файле секция **<target>** позволяет выполнить определенную задачу. Первые две задачи с именами **"Start MySQL"** и **"Stop MySQL"** запускают указанный исполняемый файл с нужными аргументами, при этом пути записаны для случая, если Eclipse

распакован в папку **eclipse**; MySQL – в папку **mysql** а рабочая область (в которой создан проект, в корне которого находится **build.xml**) вместе с двумя названными папками лежат в одной родительской папке. В секции `<target name="Generate data base">` в строке `<pathelement location="web/WEB-INF/lib/mysql-connector-java-5.1.17-bin.jar"/>` указывается расположение драйвера MySQL относительно папки проекта (если в проекте создана папка **lib** и в нее скопирован файл **Shop/web/WEB-INF/lib/mysql-connector-java-5.1.17-bin.jar**, то можно оставить эту строку без изменений. В строке `<fileset dir="sql" includes="*.sql"/>` указываются все файлы с расширением **sql** в кодировке UTF-8, содержащим SQL-запросы к базе данных (как видно в примере, JDBC-URL не содержит имени базы данных, она создается непосредственно в запросах, там же выбирается в качестве активной БД с помощью SQL-оператора USE). Данные файлы берутся из папки **sql** в папке проекта. Секция `<sort>` обеспечивает выполнение скриптов в порядке, определяемой лексико-графической сортировкой.

3. во вкладке Ant добавить созданный файл **build.xml** (например, через контекстное меню).
4. во вкладке Ant должно появиться три задачи, запуск любой из них осуществляется двойным щелчком по ней.

Лабораторная №3
Сервисы и транзакции
4 часа

1. Описать для каждой сущности интерфейс – сервис для ее обслуживания. Методы этого интерфейса во многом будут повторять методы интерфейсов DAO (см. пакет **service**).
2. Реализовать все описанные интерфейсы, в реализации использовать созданные ранее DAO. Сервисы должны решать следующие задачи:
 - восстановление взаимосвязей между объектами
 - обработка транзакций
3. Реализовать класс (классы), предоставляющие доступ к требуемой реализации

Лабораторная №4

Сервлеты и JSP

4 часа

1. Реализовать сервлет–контроллер пользовательских запросов (см. пакет **controller**).
2. Описать иерархию классов–обработчиков пользовательских запросов на основе шаблона Command (см. пакет **action**).
3. Реализовать авторизацию пользователя в системе

Репозиторий ВГУ

Лабораторная №5
Реализация функциональных требований
12 часов

1. Реализовать все функциональные требования, сформулированные для приложения.
2. Обработку пользовательских запросов организовать в специальных классах (иерархия которых была создана на предыдущей лабораторной).
3. Визуализацию данных, подготовленную этими обработчиками, осуществить в JSP-страницах. JSP-страницы не должны содержать логики приложения и какой-либо другой java-код. Для проверки наличия данных и перебора элементов коллекций использовать возможности JSTL и Expression Language.

Варианты индивидуальных проектов

1. Вариант

"Факультет"

ИС должна содержать сведения о перечне специальностей и учебных курсах для них, преподавателях, студентах, распределении преподавателей по учебным курсам, распределении студентов по специальностям.

2. Вариант

"Общежитие"

ИС должна содержать сведения о корпусах и комнатах общежития, сведения о проживающих в общежитии и их размещении по комнатам.

3. Вариант

"Нагрузка преподавателей"

ИС должна содержать сведения о преподавателях, учебных дисциплинах, кафедрах вуза, преподавательском составе кафедр, распределении учебных дисциплин между преподавателями.

4. Вариант

"Бюро Кредитных историй"

ИС должна содержать сведения о человеке, бравшем кредит, банках и организациях, выдававших кредит.

5. Вариант

"Торговая компания"

ИС должна содержать сведения о офисах компании, расположенных в разных городах, служащих компании, объемах продаж офисов за каждый день.

6. Вариант

"Приемная комиссия"

ИС должна содержать сведения об абитуриентах, расписании экзаменов, результатах экзаменов, членах экзаменационной комиссии, назначении членов экзаменационных комиссий на экзамены.

7. Вариант

"Библиотека"

ИС должна содержать сведения о читателях, книжном фонде, выдаче книг читателям, предварительных читательских заказах на литературу.

8. Вариант

"Патентный отдел"

ИС должна содержать сведения об авторах изобретений и поданных ими заявках на изобретения, выданных авторских свидетельствах.

9. Вариант

"Учебный фонд библиотеки"

ИС должна содержать сведения о хранящихся в библиотеке учебных и методических пособиях, изучаемых студенческими группами учебных дисциплинах и перечнях пособий, необходимых при изучении дисциплин, студентах академических групп и о выдаче учебных пособий студентам.

10. Вариант

"Спортивные секции"

ИС должна содержать сведения о спортивных секциях вуза, занимающихся в них студентах, расписании занятий секций, тренерах секции.

11. Вариант

"ГАИ"

ИС должна содержать сведения об автомобиле, его владельце, правонарушениях владельца и сотрудников ГАИ, зафиксировавших правонарушение.

12. Вариант

"Аэропорт"

ИС должна содержать сведения о рейсах, типе авиалайнера, пассажирах, количестве проданных и забронированных билетов.

13. Вариант

"Аспирантура"

ИС должна содержать сведения о плане приема в аспирантуру, расписании приемных экзаменов по специальностям, данные о поступающих, результатах сдачи экзаменов.

14. Вариант

"Издательство"

ИС должна содержать сведения о книгах, авторах, сроках издания книги, расходных материалах на издание, оптовых покупателях изданий и о количестве проданных книг покупателями.

15. Вариант

"Телефонная компания"

ИС должна содержать сведения о клиентах, предоставляемых им услугах, тарифных планах.

16. Вариант

"Туристическое агентство"

ИС должна содержать сведения о туристических маршрутах, гостиницах, программе отдыха.

17. Вариант

"Агентство недвижимости"

ИС должна содержать сведения об агентстве, о его клиентах, о продаваемых и покупаемых квартирах и иных площадях.

18. Вариант

"Прокат"

ИС должна содержать сведения о товаре, выданном в прокат, человеке, взявшем товар в прокат, сроках проката, стоимости проката

19. Вариант

"Железнодорожная касса"

Пассажир делает запрос на станцию назначения, время и дату поездки. Система осуществляет поиск подходящих поездов. Пассажир выбирает поезд и бронирует билет. Кассир выдает забронированные билеты, фиксируя продажу.

20. Вариант

«Автобаза»

Диспетчер распределяет маршрутные листы на рейсы между водителями и назначает для этого автомобиль. Водитель может сделать заявку на ремонт. Водитель делает отметку о выполнении рейса.

21. Вариант

"Успеваемость"

ИС должна содержать сведения о студентах, студенческих группах, учебных дисциплинах, составе студенческих групп и расписании экзаменов, успеваемости студентов.

22. Вариант

"Поликлиника"

ИС должна содержать сведения о пациентах, врачах, расписании работы врачей и посещение пациентами поликлиники, заказах талонов.