

Лабораторная работа №3

Реализация шаблона проектирования Data Access Object

Цель

Получить навыки практического использования средства JDBC API для реализации объектов доступа к данным (DAO).

Задание

1. Разработать не менее двух Java-классов, представляющих модель данных приложения.
2. Разработать Java-классы, реализующие шаблон проектирования Data Access Objects (DAO) и обеспечивающие независимость бизнес-логики от механизма хранения данных.
3. Для каждого класса модели данных в объекте доступа к данным должны быть предусмотрены операции сохранения, изменения, удаления, извлечения экземпляра по идентификатору, извлечения всех экземпляров (если это имеет смысл в предметной области), а также методы поиска экземпляров по различным критериям.
4. Методы поиска и извлечения всех экземпляров должны возвращать коллекции объектов предметной области.
5. Разработать тесты для объекта(-ов) доступа к данным и выполнить тестирование. В результате все тесты должны быть пройдены успешно.

Вопросы для допуска

1. Какие классы модели данных будут разработаны, каким реляционным таблицам они соответствуют?
2. Какие методы поиска объектов предметной области будут реализованы в объекте доступа к данным?
3. Каким образом в объекте доступа к данным будет организовано получение и разрыв соединения с базой данных?

Теоретические сведения

Шаблон проектирования Объект доступа к данным (Data Access Object – DAO) используется для инкапсуляции доступа к источнику данных. Части шаблона и связи между ними приведены на рис. 1.

Бизнес-объект (BusinessObject) реализует бизнес-логику приложения и является клиентом объекта доступа к данным.

Бизнес-объект выполняет действия над объектами предметной области (ValueObject), получаемыми от объекта доступа к данным.

Объект доступа к данным (DataAccessObject) полностью скрывает реализацию источника данных от своих клиентов, что позволяет изменять механизм хранения данных, не затрагивая клиентов.

В качестве источника данных (DataSource) может выступать файл, реляционная или объектная база данных, LDAP-каталог, а также какая-либо внешняя система, доступ к которой ведется с использованием сокетов, веб-сервисов или любой другой технологии.

Если при работе с источником данных возникают исключения, то объекта доступа к данным «заворачивает» их в особые исключения доступа к данным (DataAccessException),

позволяя бизнес-объекту отреагировать на ошибку независимым от источника данных способом.

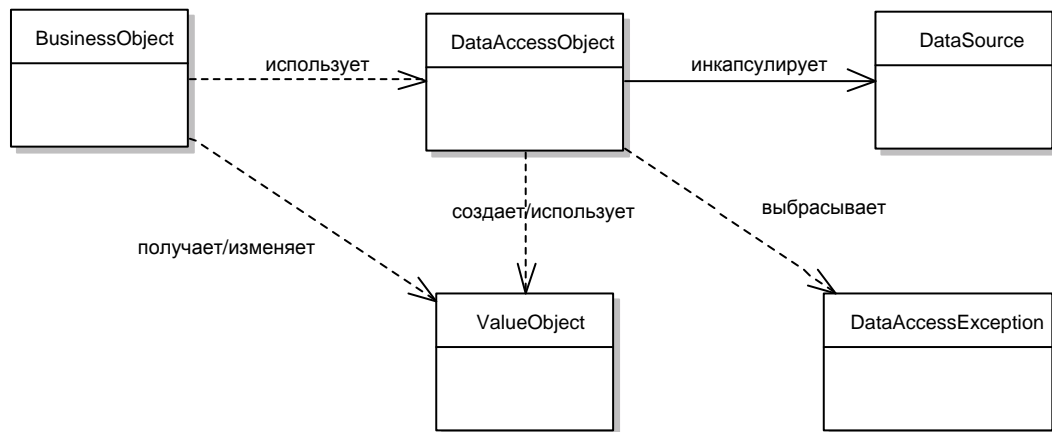


Рис. 1. Шаблон проектирования Объект доступа к данным

На рис. 2 приведена диаграмма последовательности, иллюстрирующая взаимодействие участников шаблона DAO

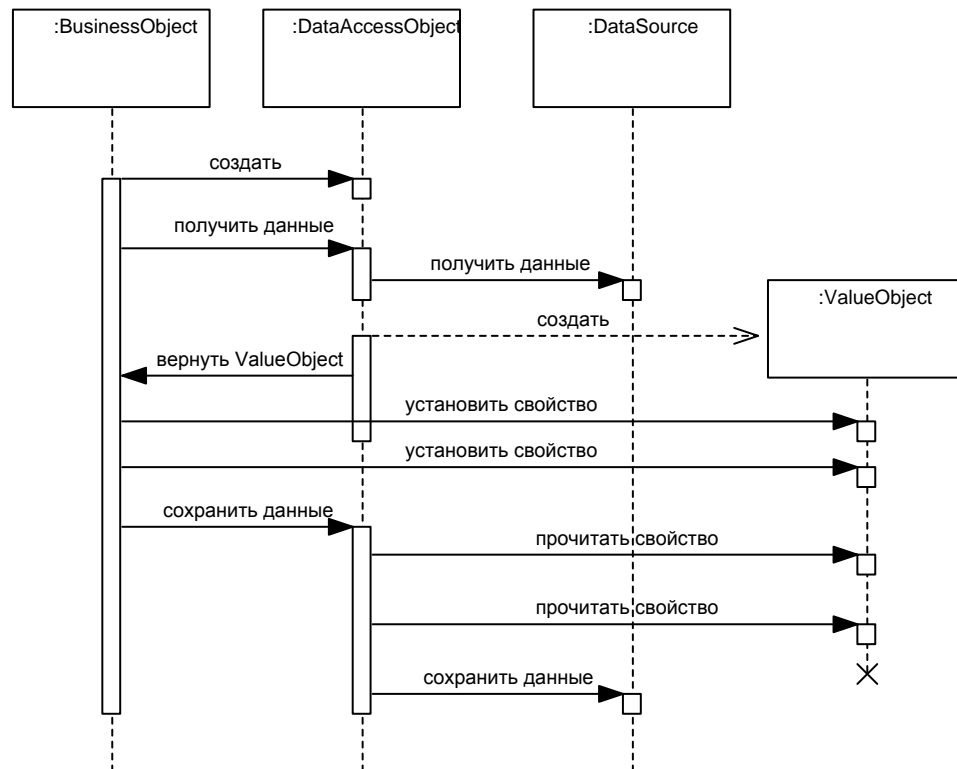


Рис. 2. Диаграмма последовательностей для шаблона проектирования Объект доступа к данным

Использование шаблона проектирования Объект доступа к данным, с одной стороны, улучшает сопровождаемость и гибкость программной системы, но с другой стороны, в некоторой мере снижает производительность из-за создания дополнительных объектов и выполнения дополнительных вызовов между слоями.

Реализацию данного шаблона можно выполнить различными способами, в зависимости от сложности модели предметной области и необходимой степени гибкости программной системы.

Порядок выполнения работы

1. Создайте классы, представляющие модель данных приложения. Желательно разместить их в отдельном пакете, который можно создать явно командой **New / Java Package...** или неявно при создании первого класса пакета. Для создания классов предназначен пункт меню **New / Java Class...**, при выборе которого выводится диалог создания класса (рис. 3).

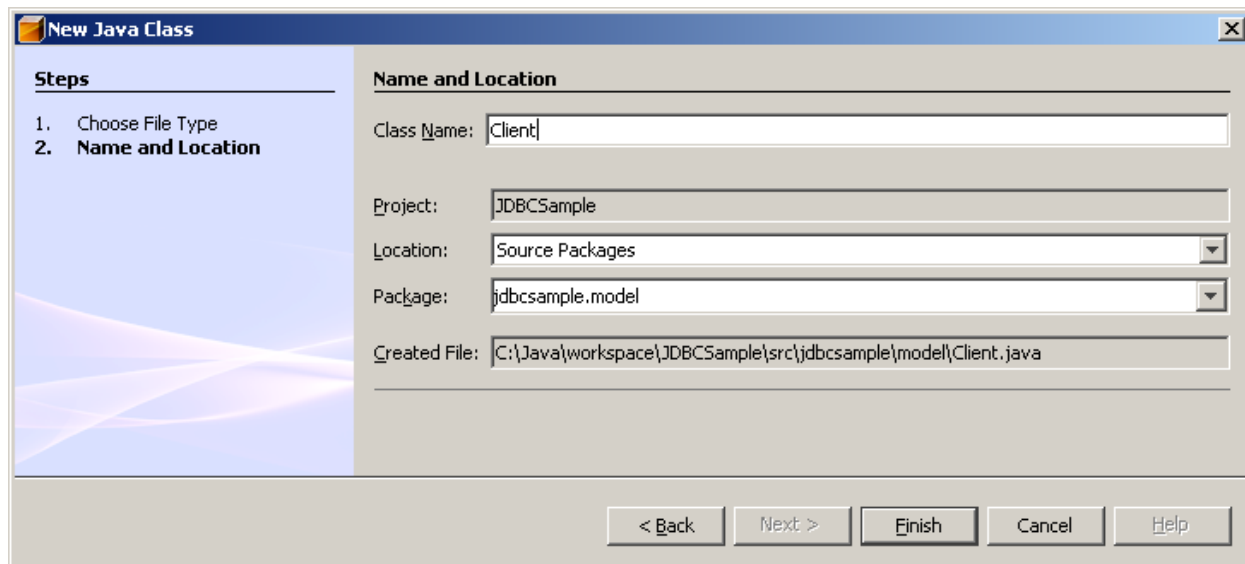


Рис. 3. Диалог создания класса

2. Класс модели данных должен содержать поля, соответствующие столбцам некоторой таблицы базы данных, и методы для считывания и изменения значений этих полей. Создать get-/set-методы для полей класса можно с помощью мастера инкапсуляции полей **Encapsulate Fields** (рис. 4), вызываемого через пункт меню **Refactor / Encapsulate Fields...**. В данном диалоге установите флажки в столбцах **Create Getter** (Создать get-метод) и **Create Setter** (Создать set-метод) для всех созданных полей, уберите флажок **Preview All Changes** и нажмите кнопку **Next >**.

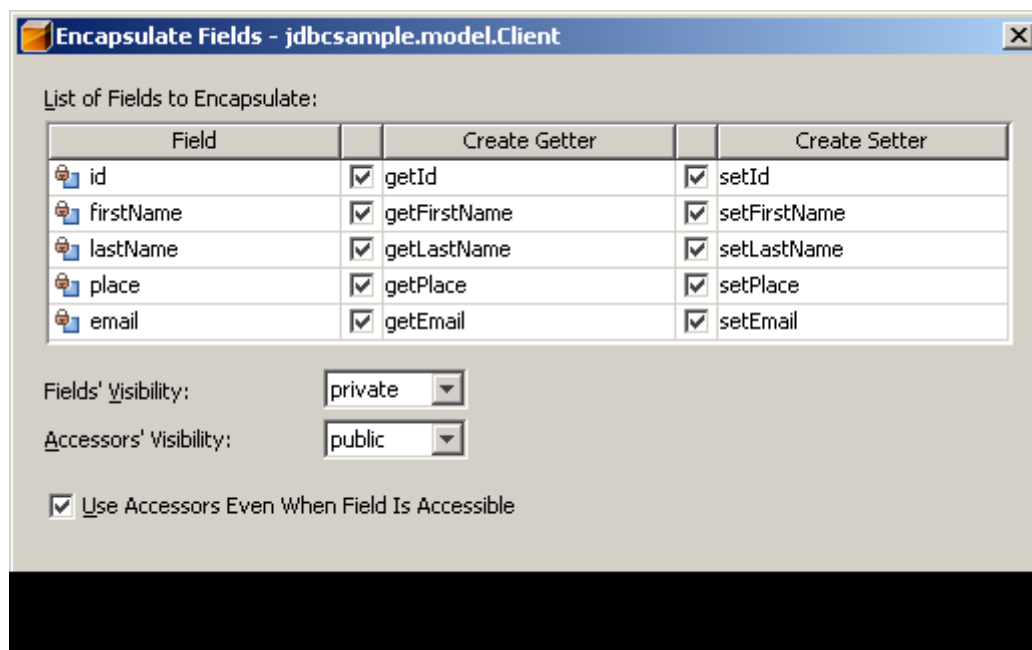


Рис. 4. Мастер инкапсуляции полей

3. Создайте классы объектов доступа к данным. Допускается определить единственный класс объекта доступа к данным для операций со всеми классами модели данных. Разработайте методы для доступа к данным, список которых приведен в задании. В качестве типов параметров и возвращаемых значений данных методов используйте разработанные классы модели данных
4. Разработайте тесты для классов объектов доступа к данным. Тестирование будет выполняться с помощью библиотеки JUnit, обеспечивающей автоматизацию процесса тестирования. В NetBeans имеется мастер создания теста для существующего класса, для вызова которого в окне создания нового файла нужно выбрать тип файла **JUnit / Test for Existing Class**. В мастере создания теста для существующего класса (рис. 5) нужно выбрать тестируемый класс в вызываемом кнопкой **Browse...** диалоге (рис. 6). Настройки генерируемого теста можно оставить по умолчанию или заменить на приведенные на рис. 5.
5. Запустите тест с помощью пункта главного меню **Run / Test “<project name>”**. При этом в нижней части окна NetBeans открывается окно **JUnit Test Results** с результатами тестирования (рис. 7). В случае возникновения исключения в данном окне можно узнать место его возникновения. Подробный отчет о запуске теста содержится в окне **Output**, располагающемся по соседству с окном JUnit Test Results. Убедитесь в успешном выполнении операций с базой данных с помощью утилиты MySQL Query Browser.

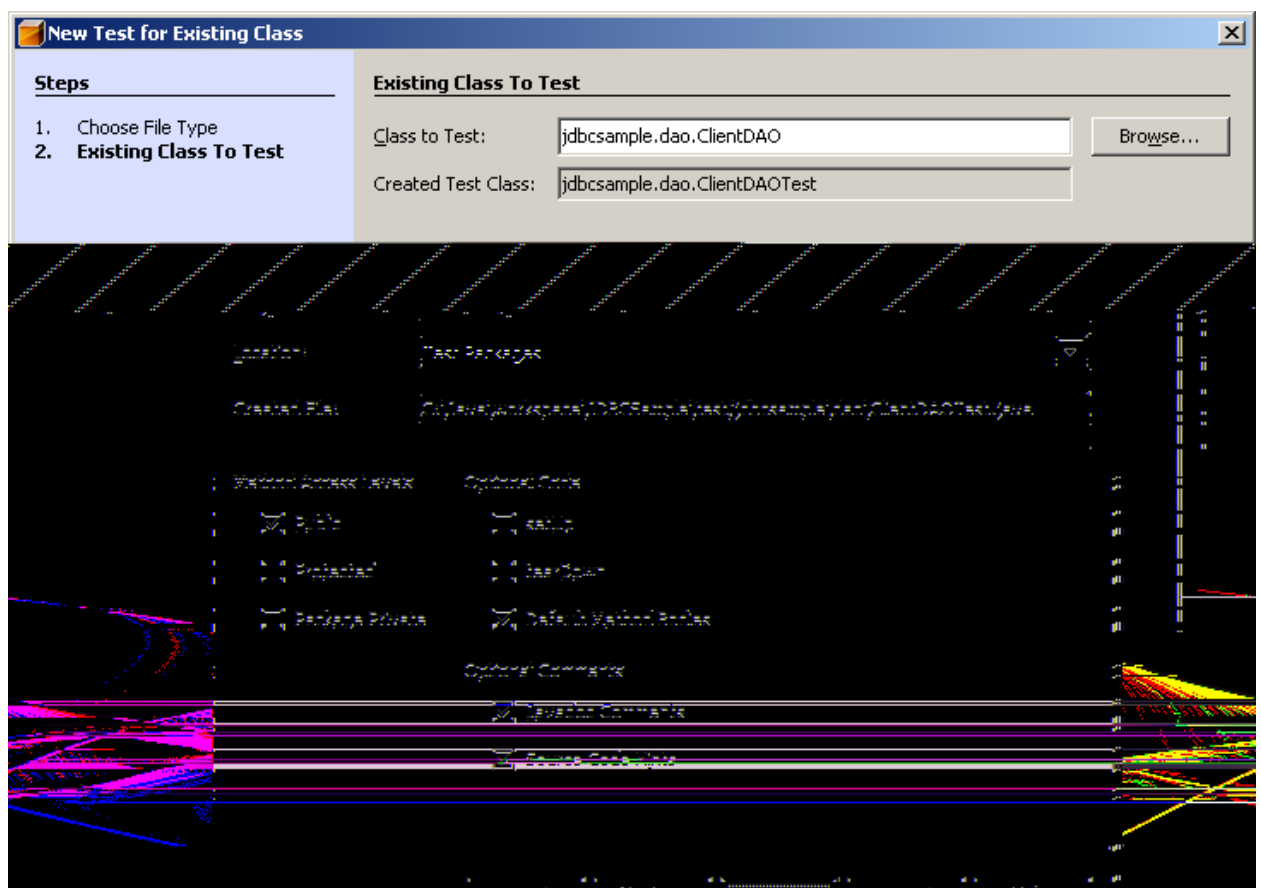


Рис. 5. Создание теста для существующего класса

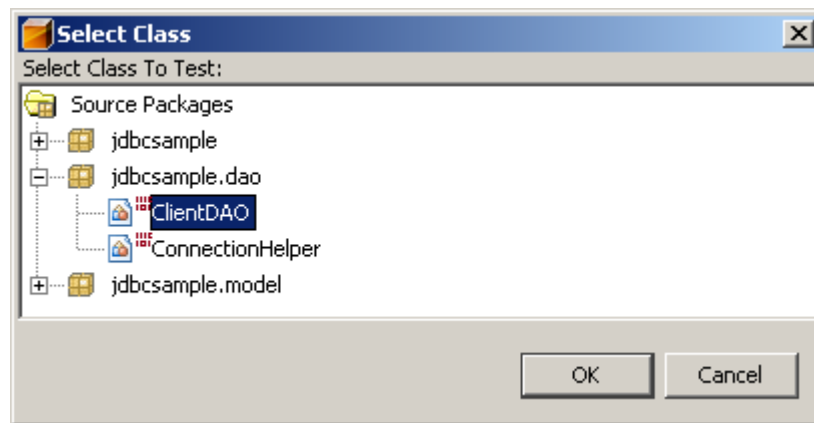


Рис. 6. Выбор класса для создания теста

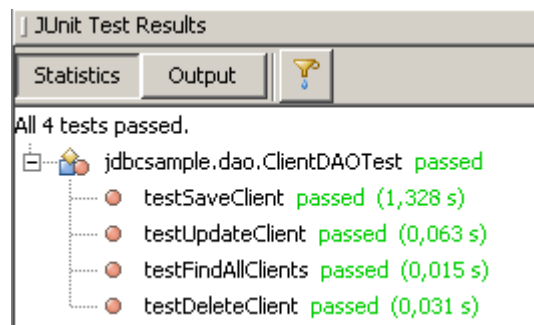


Рис. 7. Результаты тестирования

Контрольные вопросы

1. Укажите соответствие между разработанными классами и элементами шаблона DAO.
2. Для чего предназначен класс исключения доступа к данным?
3. Какие функции помимо переноса данных между базой данных и бизнес-логикой приложения можно возложить на объекты предметной области?
4. Каким образом устанавливается соответствие между полями объектов предметной области и столбцами реляционных таблиц?