Федеральное государственное бюджетное образовательное учреждение   
высшего образования

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | «Московский государственный технический университет  им. Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана) |

ФАКУЛЬТЕТ – Информатика и управления

КАФЕДРА – Информационные системы и телекоммуникации

РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе на тему

Разработка модели данных диалога для HyperGraphDB

Студент группы ИУ3-71 (подпись) 20.01.2017 А.Э.Айрапетов

Руководитель курсовой работы (подпись) 20.01.2017 А.М. Иванов

Москва, 2017

HypergraphDB[1] является графовой базой данных общего названия с открытым исходным кодом. То есть сетевая модель данных в такой базе данных представляет собой граф. Узлы такого графа представляют собой Java-объекты (HypergraphDB реализован на Java).

Модель данных диалога подразумевает наличие класса сообщения, класса пользователя (отправителя) и класса самого диалога.

Ниже приведен листинг данных классов.

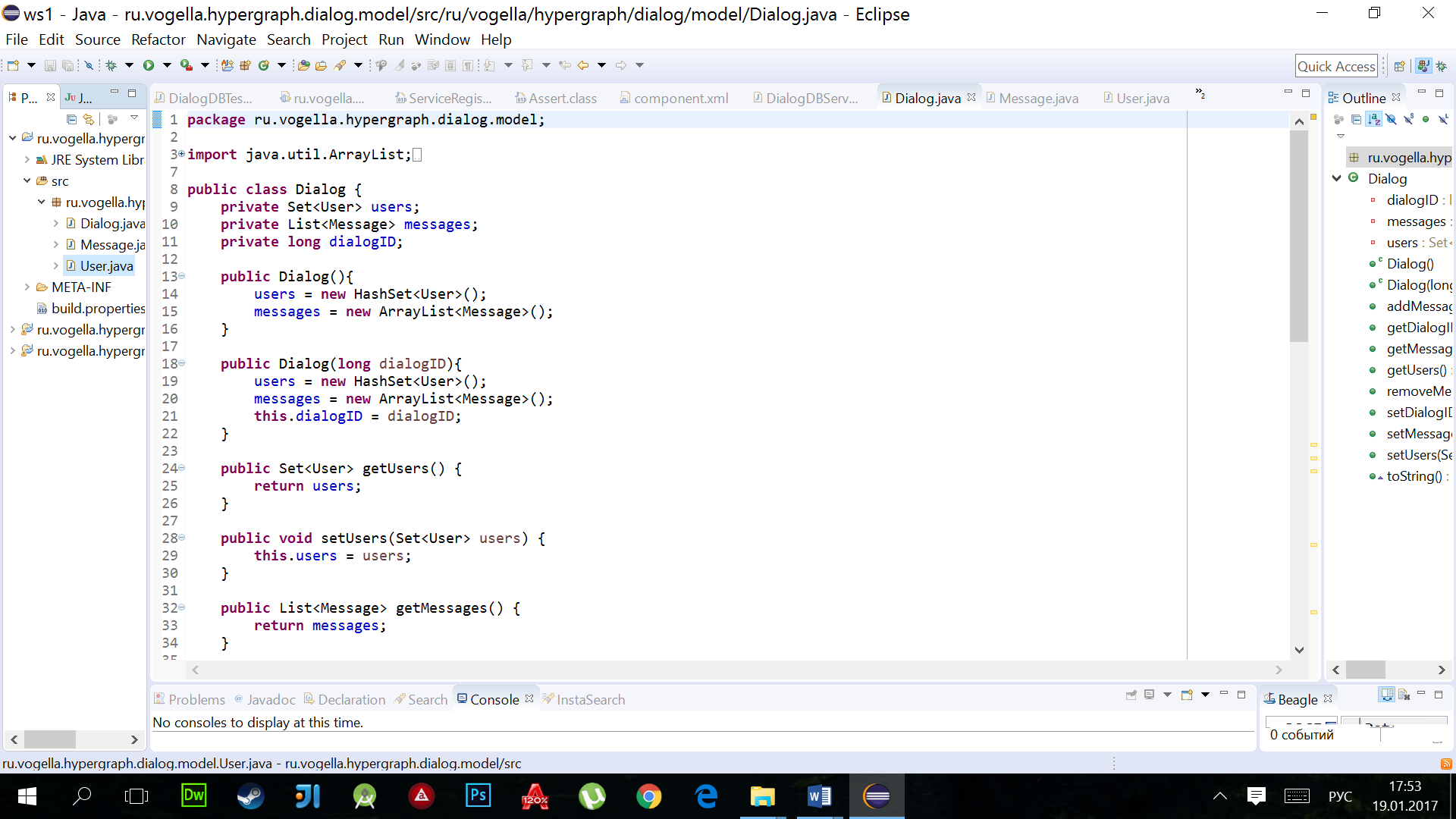


Рисунок 1 – поля и конструкторы класса Dialog

Объект класса Dialog содержит список ссылок на объекты класса User – участников диалога, список ссылок на объекты типа Message – сообщений диалога, а также поле, содержащее id диалога.

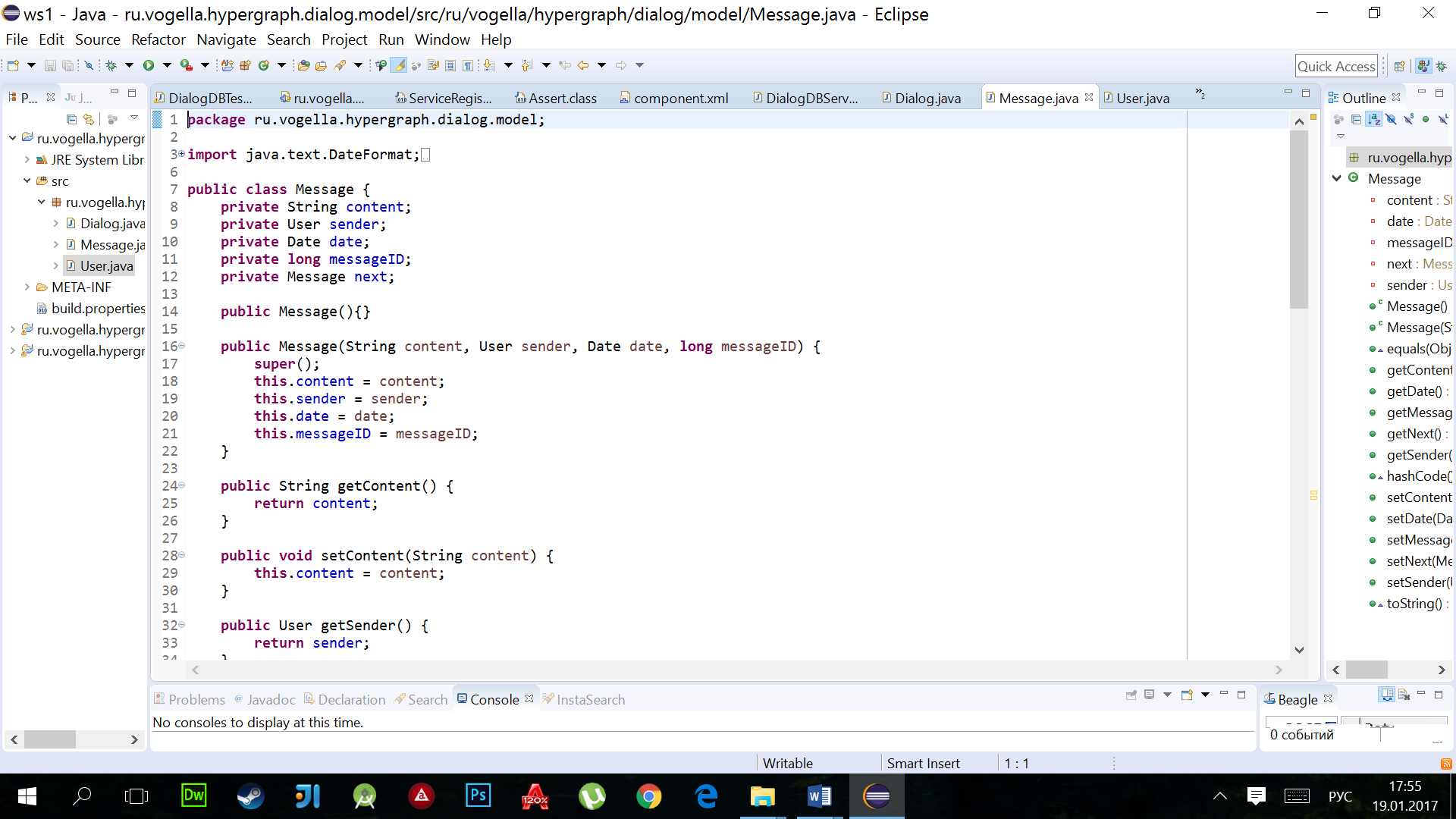


Рисунок 2 – поля и конструкторы класса Message

Объект класса Message содержит всю информацию о данном сообщении, а также ссылку на объект следующего сообщения, если такой существует.

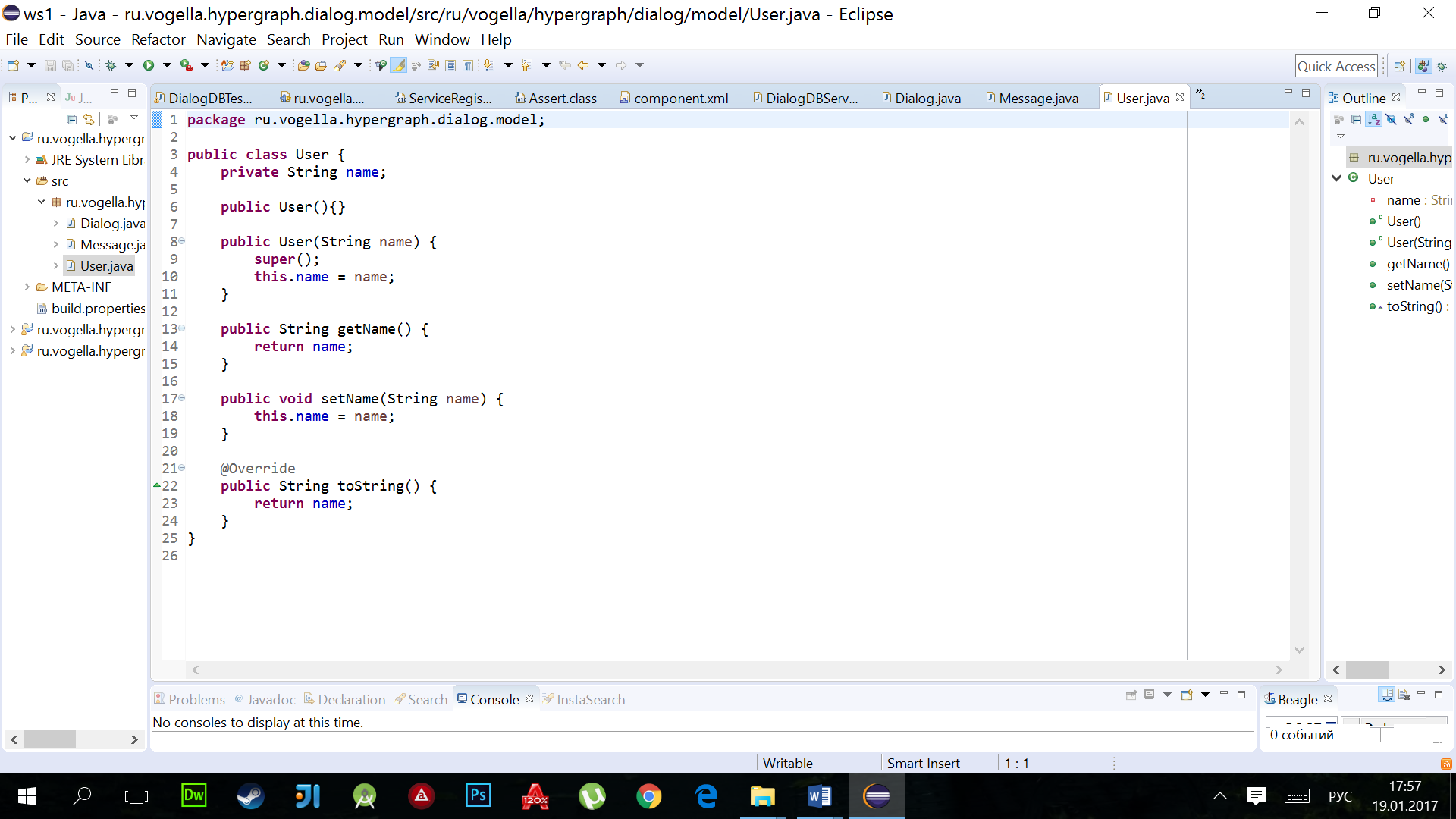


Рисунок 3 – поля и конструкторы класса User

Объект класса User содержит имя пользователя.

Для того, чтобы объекты данных классов могли быть использованы в качестве узлов графа в HypergraphDB, они должны иметь пустой конструктор (не содержащий аргументов) и геттеры и сеттеры для всех своих полей.

Объект класса Dialog содержит ссылки на все сообщения в этом диалоге. Объект класса Message содержит ссылку на объект следующего сообщения, если такой существует. Программа была реализована таким образом, чтобы минимизировать работу с самой HypergraphDB и работать с этими объектами по принципам объектно-ориентированного программирования.

Однако появляется необходимость в методе, который преобразовывал бы данные объекты (содержащие ссылки на другие объекты и т.д.) в граф, то есть записывал бы информацию в графовую базу данных, а также в метод, который бы наоборот преобразовывал граф в Java-объекты.

Модель графа, в который должен преобразовываться объект класса Dialog приведена на рисунке 4.

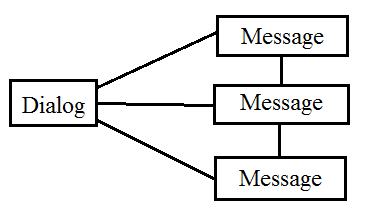


Рисунок 4 – модель графа диалога

Так как с помощью HypergraphDB мы описываем графовую базу данных, то появляется необходимость создавать не только узлы графа, но и ребра графа. В данной модели данных на ребра графа заменяются ссылки на объекты внутри других объектов при записи объектов Java-кода в базу данных Hypergraph.

Таким образом, был реализован сервис, позволяющий выполнять описанные «переходы» от Java-объектов к графам.

Сервис содержит 4 метода: для добавления и удаления диалога, для получения сообщения по id и для получения диалога по id.

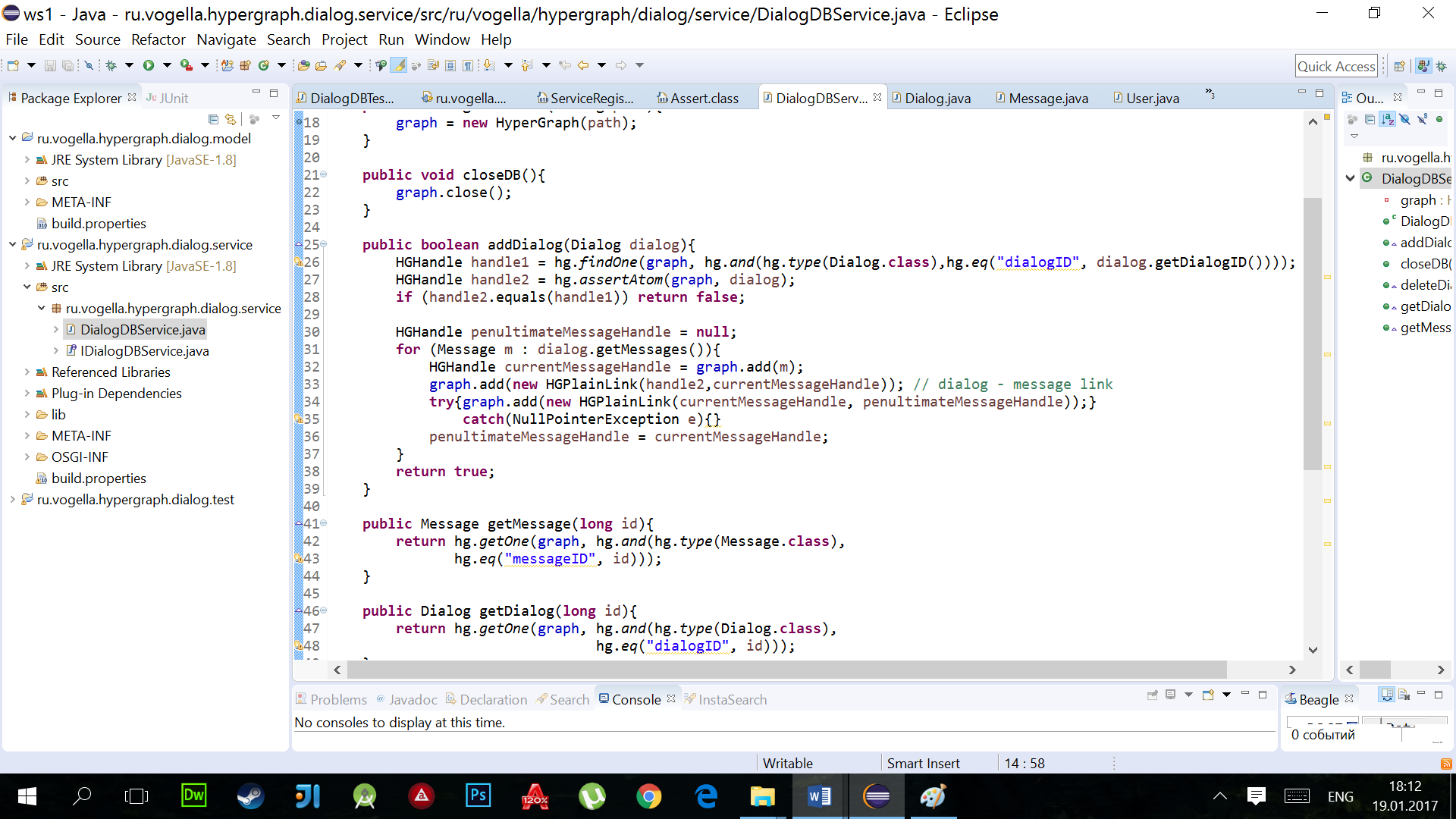


Рисунок 5 – метод добавления диалога в базу данных

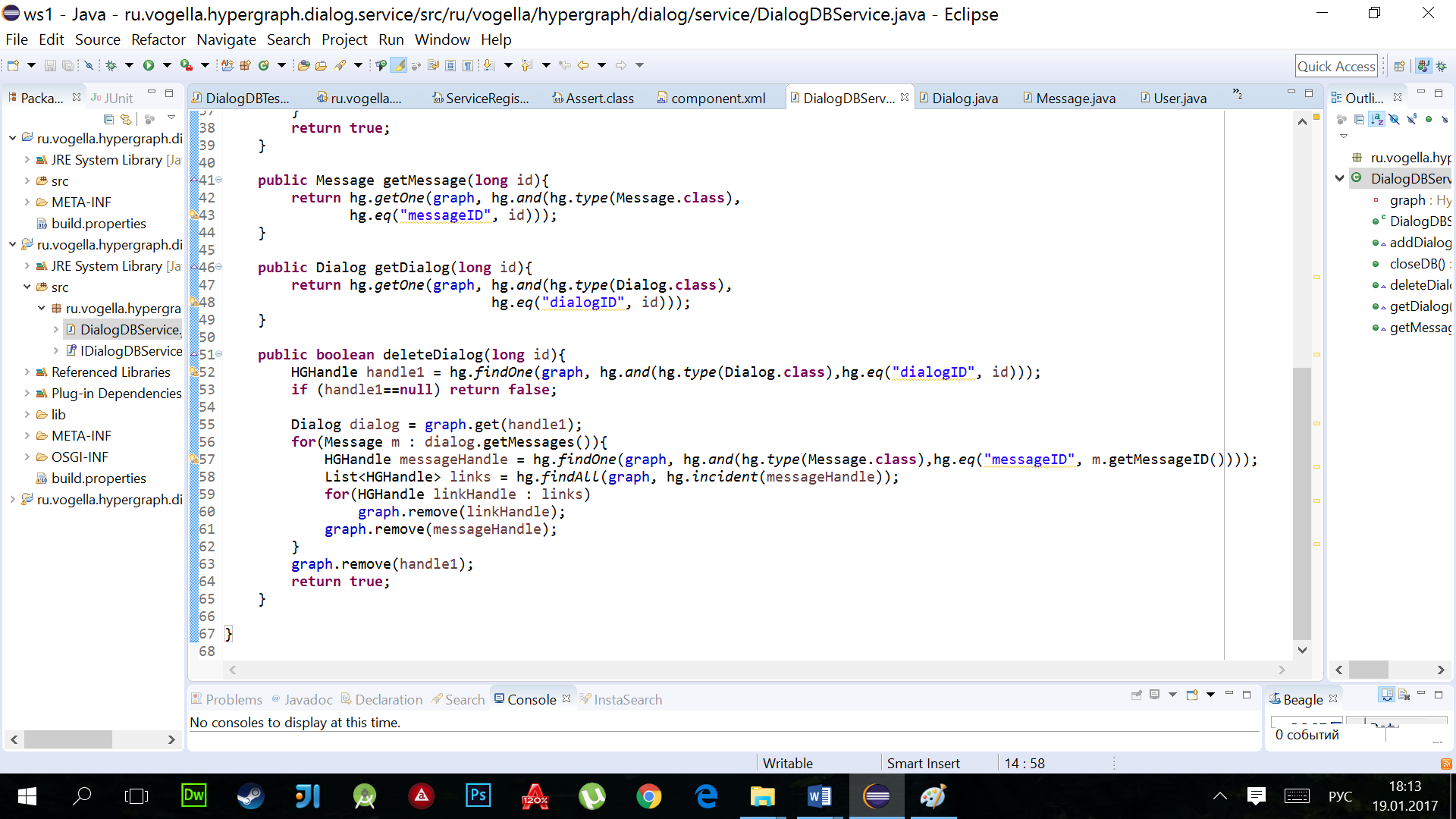


Рисунок 6 – метод удаления диалога из базы данных

Удаляются не только узлы графа, но и его ребра.

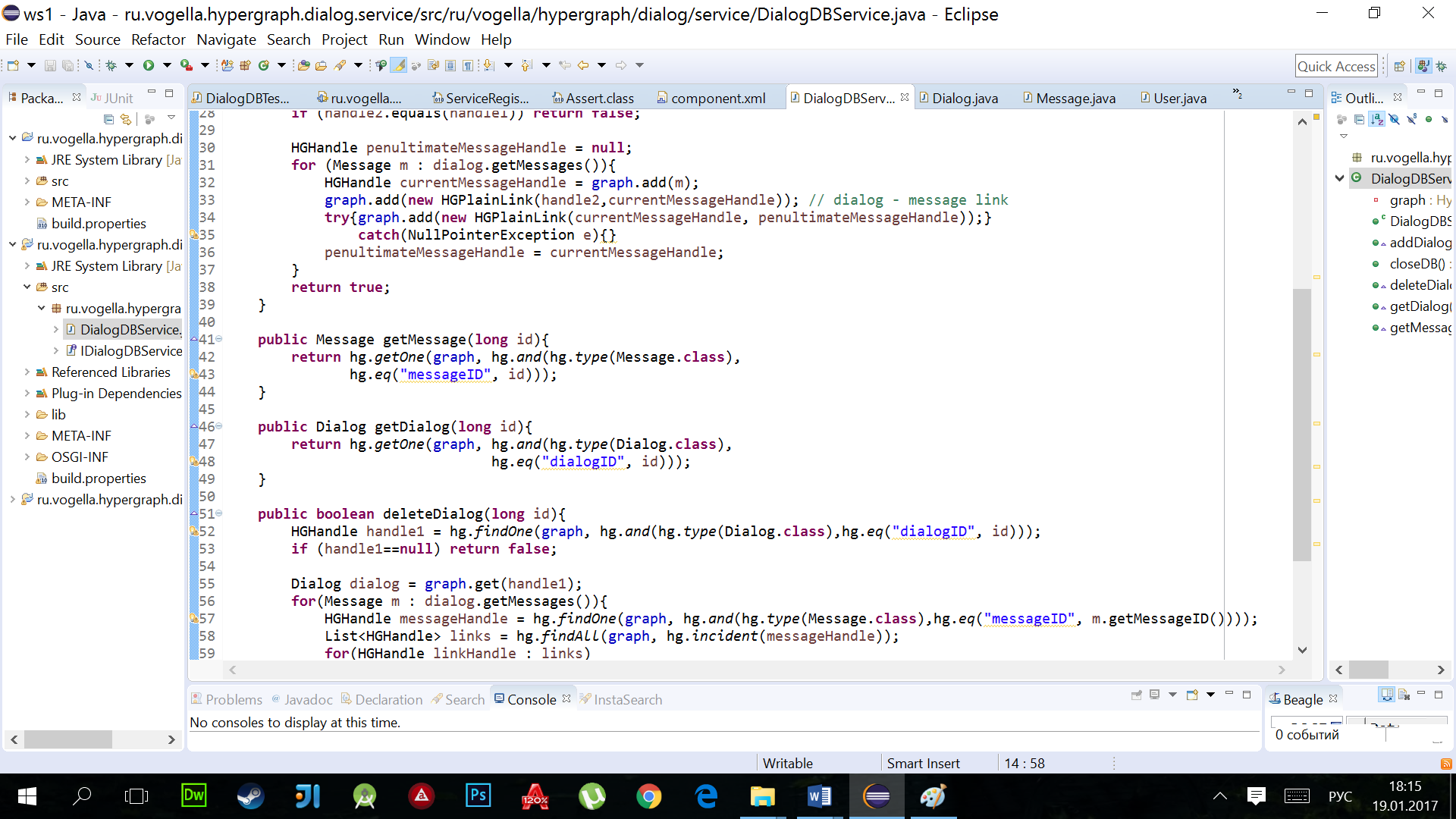


Рисунок 7 – метод получения сообщения по id

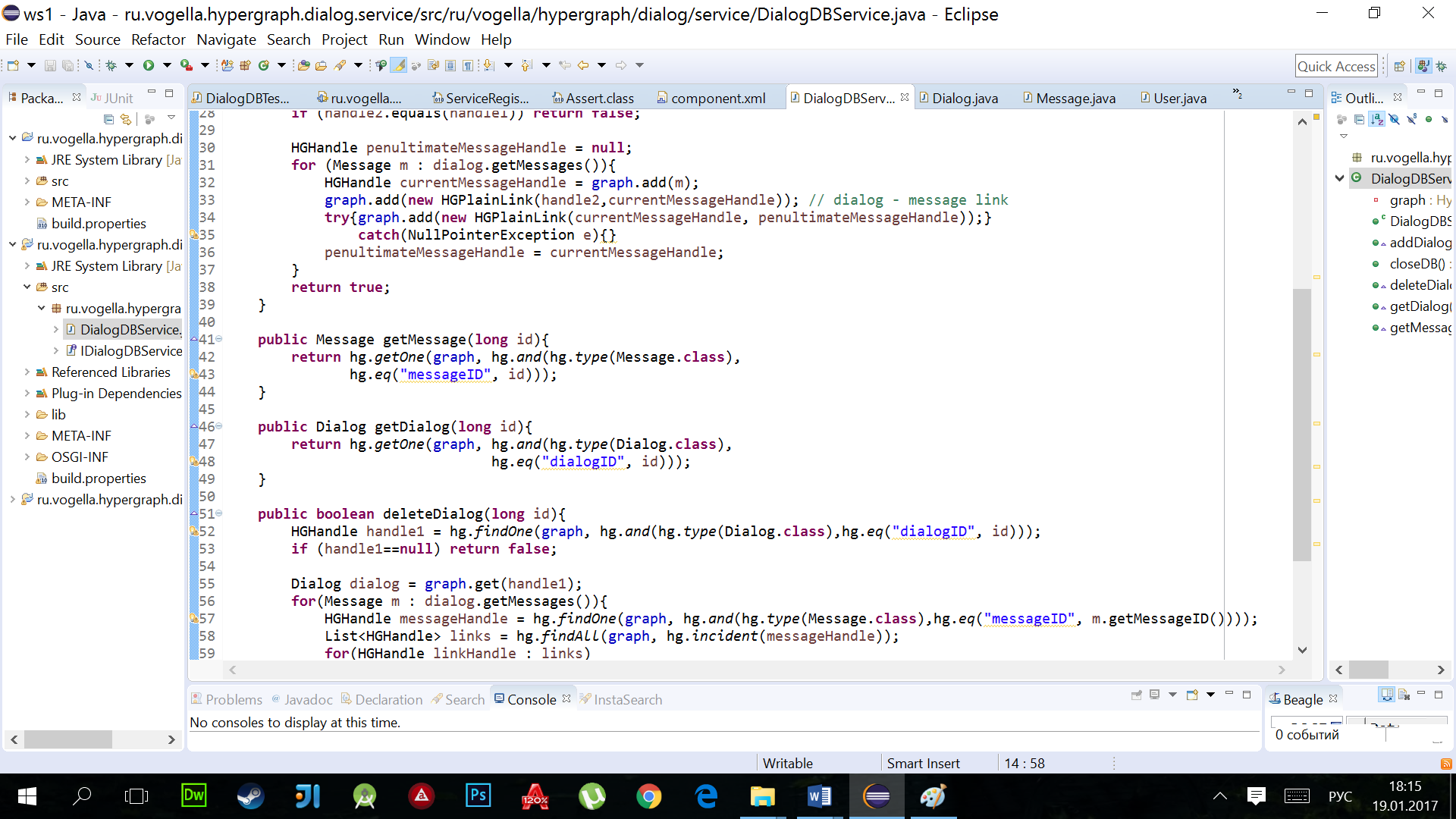


Рисунок 8 – метод получения диалога по id

Для контроля правильности работы данного сервиса были разработаны юнит-тесты, использующие для обеспечения модульного тестирования библиотеку JUnit.

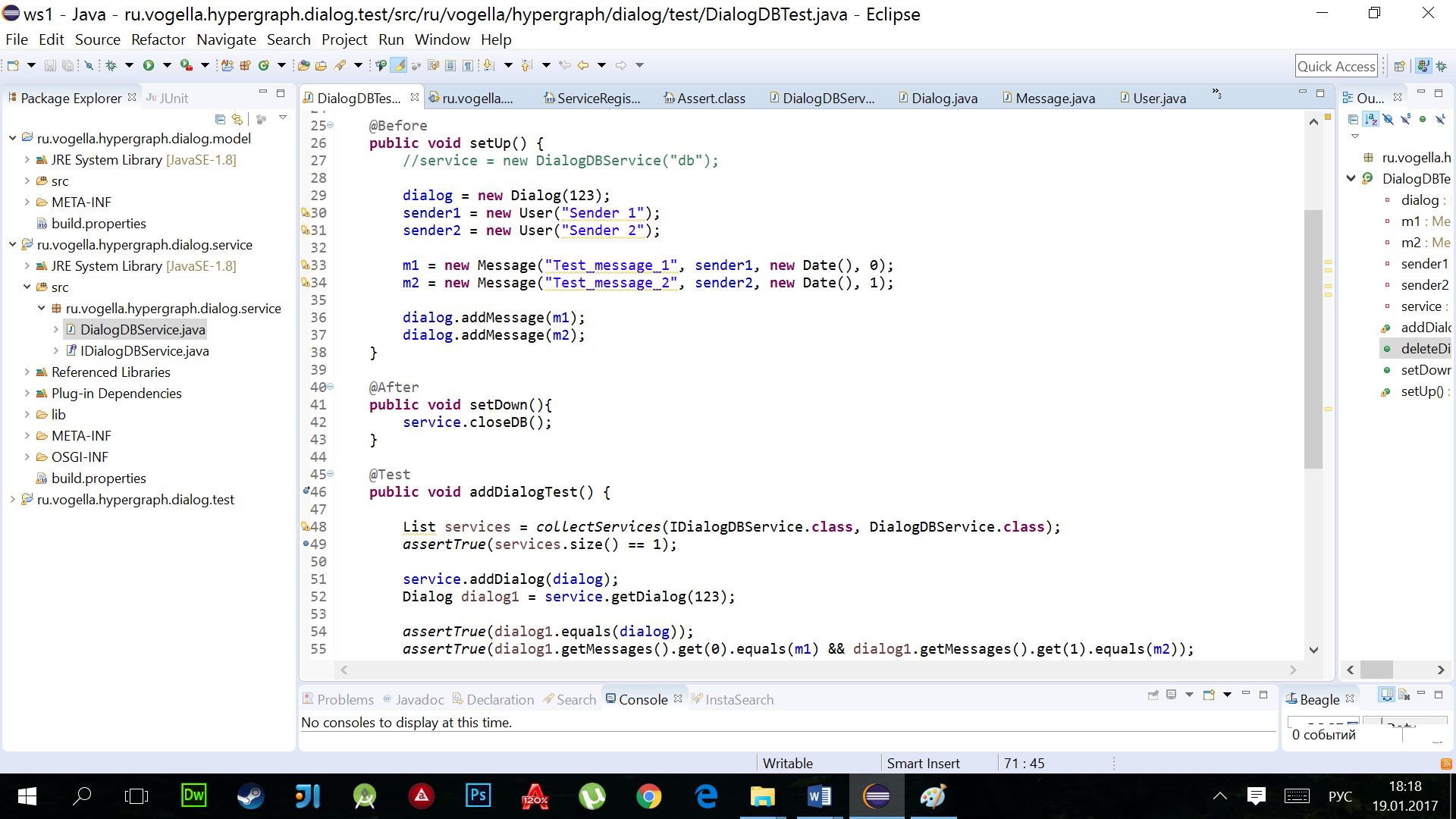


Рисунок 9 – данные методы выполняются до и после выполнения каждого юнит-теста соответственно

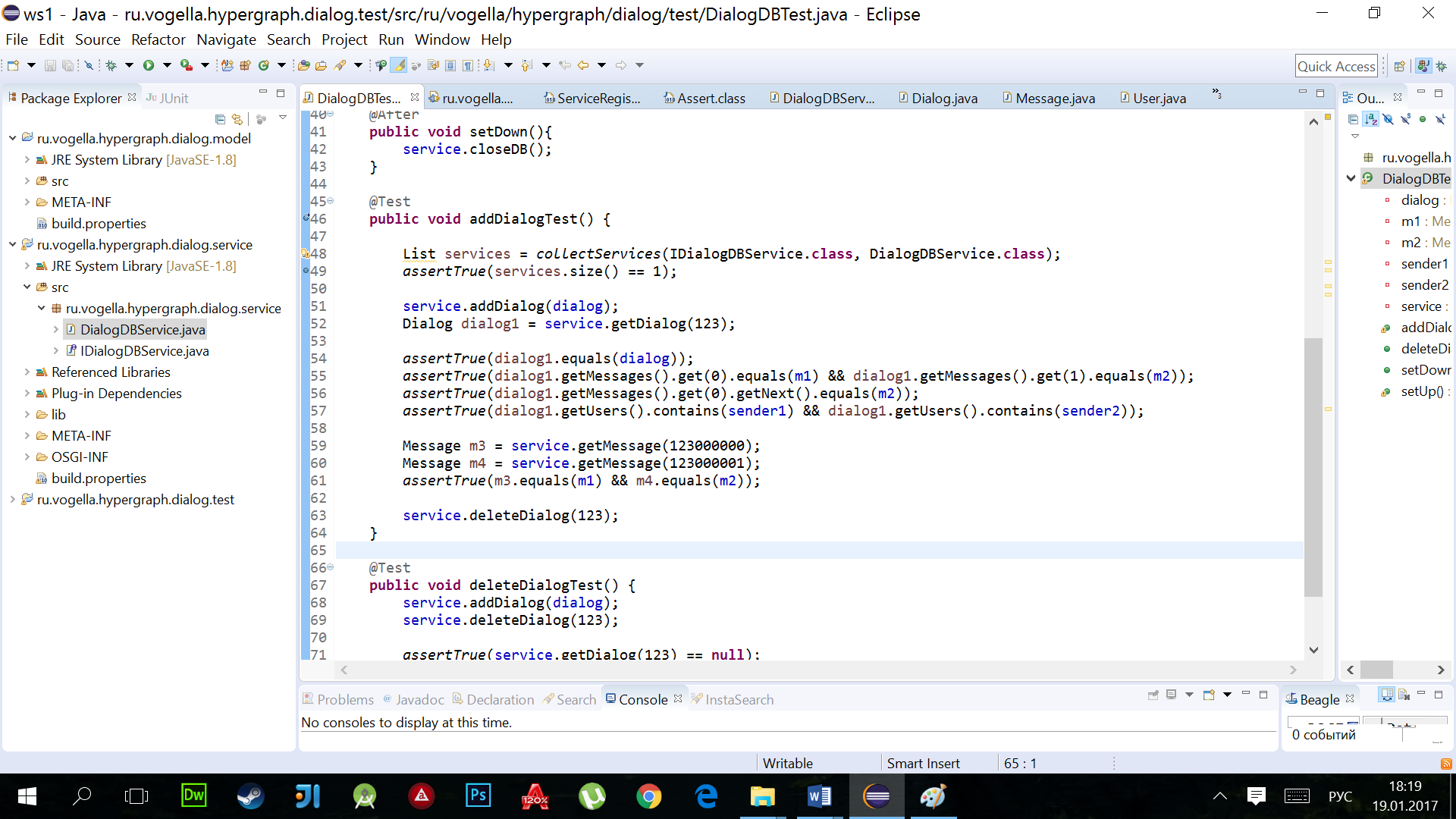


Рисунок 10 – тест метода добавления диалога в базу данных

Данный юнит-тест проверяет корректность записи объекта диалога в базу данных – остается ли объект таким же при хранении в качестве узла графа, содержит ли ссылки на те же объекты, содержат ли сообщения диалога ссылки на следующие сообщения.

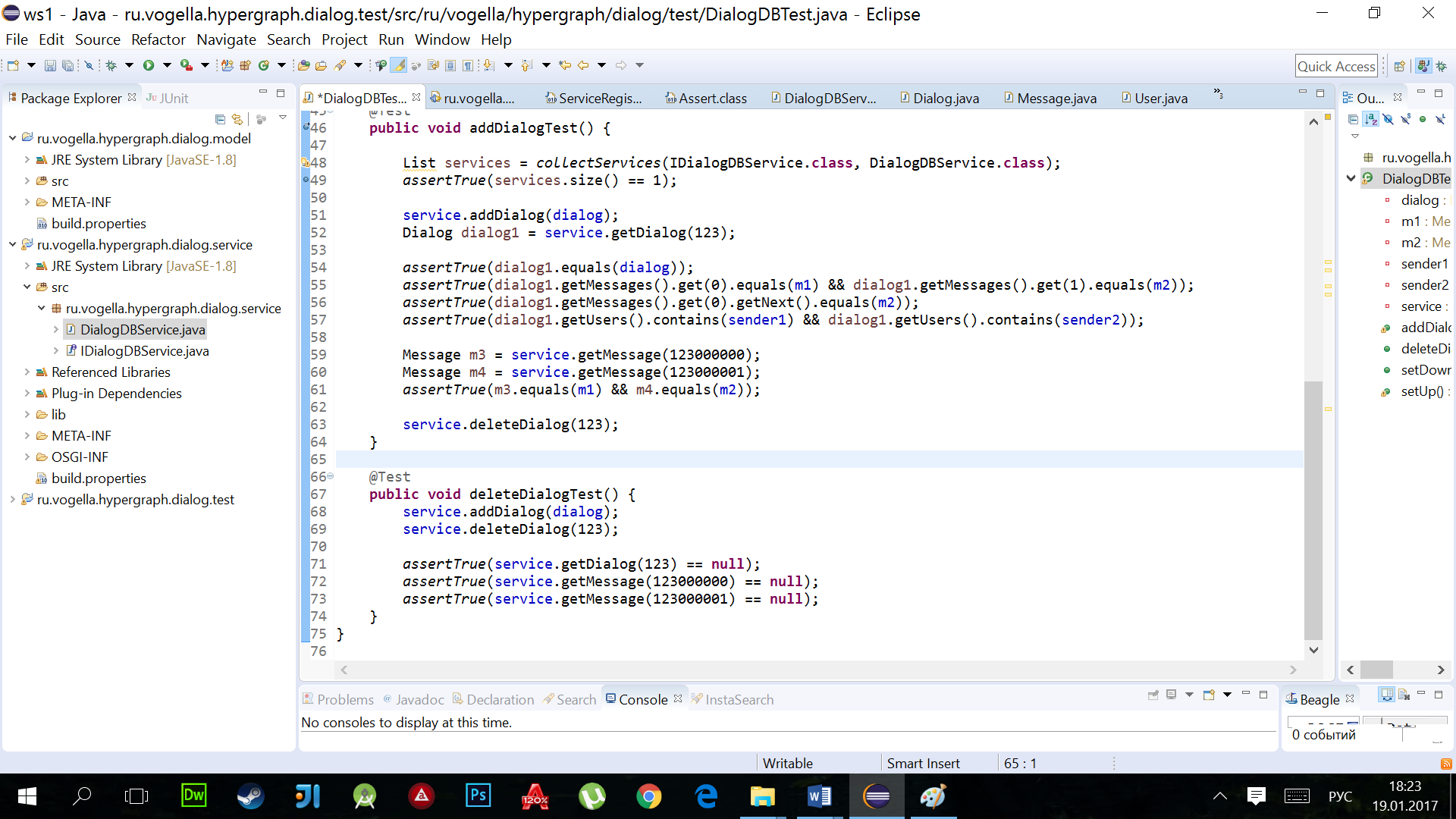


Рисунок 11 – тест метода удаления диалога из базы данных

Данный юнит-тест проверяет что диалог, а также все его сообщения корректно удаляются из базы данных.

Список литературы

[1] HypergraphDB – сайт проекта: [Электронный ресурс]. – Режим доступа: www.hypergraphdb.org. Дата обращения –15.01.2017.