

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Кафедра Систем информатики

Направление подготовки 09.06.01 – Информатика и вычислительная техника

ОТЧЕТ

Обучающегося Петров Владимир Артемович группы № 22215 курса 4
(Ф.И.О. полностью)

Тема задания: Реализация репозитория онтологии для графовой базы данных
Neo4j

Новосибирск 2025

Оглавление

1. Инициализация подключения
2. Закрытие подключения
3. Операции с классами
 - 3.1. Создание класса
 - 3.2. Получение класса
 - 3.3. Обновление класса
 - 3.4. Удаление класса
 - 3.5. Получение родителей класса
 - 3.6. Получение дочерних классов
4. Операции с объектами
 - 6.1. Создание объекта
 - 6.2. Получение объекта
 - 6.3. Обновление объекта
 - 6.4. Удаление объекта
5. Атрибуты классов
 - 7.1. Добавление DatatypeProperty
 - 7.2. Удаление DatatypeProperty
 - 7.3. Добавление ObjectProperty
 - 7.4. Сбор сигнатуры класса
6. Произвольные запросы
7. Заключение

1. Инициализация подключения

`__init__(self, uri, user, password)`

Назначение: устанавливает соединение с базой данных Neo4j.

Параметры:

- `uri` — строка подключения;
- `user` — имя пользователя;
- `password` — пароль пользователя.

2. Заккрытие подключения

`close(self)`

Назначение: корректно завершает соединение с базой данных, освобождая ресурсы.

3. Операции с классами

3.1. `create_class(title, description, parent_uri=None)`

Создаёт узел с меткой `Class`. Устанавливает уникальный `uri`, `title`, `description`.

3.2. `get_class(uri)`

Возвращает узел класса по URI.

3.3. `update_class(uri, title=None, description=None)`

Обновляет свойства класса.

3.4. `delete_class(uri)`

Удаляет узел класса и все связанные элементы.

3.5. `get_class_parents(uri)`

Возвращает список родительских классов.

3.6. get_class_children(uri)

Возвращает список дочерних классов.

4. Операции с объектами

4.1. create_object(class_uri, title, description)

Создаёт узел с меткой Object, принадлежащий заданному классу.

4.2. get_object(uri)

Возвращает объект по URI.

4.3. update_object(uri, title=None, description=None)

Обновляет свойства объекта.

4.4. delete_object(uri)

Удаляет объект и все его связи.

5. Атрибуты классов

5.1. add_class_attribute(class_uri, title)

Создаёт узел DatatypeProperty и связь с классом.

5.2. delete_class_attribute(class_uri, title)

Удаляет DatatypeProperty с указанным названием.

5.3. add_class_object_attribute(class_uri, title, target_class_uri)

Создаёт узел ObjectProperty и связь между двумя классами.

5.4. collect_signature(class_uri)

Возвращает структуру с двумя списками:

- datatype_properties
- object_properties

6. Произвольные запросы

run_custom_query(query, params=None)

Позволяет выполнять любые Cypher-запросы к базе данных.

Пример:

```
repo.run_custom_query("MATCH (n:Class) RETURN count(n)")
```

7. Заключение

Класс **OntologyRepository** реализует объектный слой для работы с графовой базой данных Neo4j в терминах онтологии.

Поддерживаются операции создания, обновления, удаления и поиска классов и объектов, а также управление их атрибутами (DatatypeProperty, ObjectProperty).