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: **Разработка драйвера для графовой базы данных Neo4j**

Новосибирск 2025

Оглавление

1. Инициализация

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

Назначение:
Устанавливает соединение с базой данных Neo4j.

Параметры:

- `uri` - строка подключения
- `user` - имя пользователя
- `password` - пароль пользователя

2. Закрытие подключения

`close(self)`

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

3. Генерация уникального URI

`generate_random_string(length=20)`

Назначение:
Создаёт уникальный идентификатор (URI) на основе UUID.

Пример использования:
Для установки `uri` при создании нового узла.

4. Вспомогательные методы

4.1. `_extract_node(node_obj: Node)`

Назначение:
Преобразует объект `Node` в Python-словарь с полями:

- `id`
- `uri`
- `labels`
- `properties`

4.2. `_extract_arc(relationship: Relationship)`

Назначение:
Преобразует объект `Relationship` в словарь с полями:

- `id`
- `uri`
- `node_uri_from`
- `node_uri_to`

4.3. `transform_labels(labels, separator=':')`

Назначение:
Форматирует список меток в строку, совместимую с синтаксисом Cypher.

Пример:

`["Person", "Student"]` → `:Person:Student`

4.4. `transform_props(props)`

Назначение:
Преобразует словарь свойств в строку формата `{key: value}` для вставки в Cypher-запрос.

Пример:

`{"name": "John", "age": 25}` → `{"name":"John", "age":25}`

5. Операции с узлами

5.1. `create_node(self, labels, properties)`

Назначение:
Создаёт новый узел с указанными метками и свойствами.

Действия:

Добавляется уникальное поле `uri`

Формируется запрос:
`CREATE (a:Label1:Label2 $props) RETURN a`

5.2. `get_nodes_by_labels(self, labels)`

Назначение:
Извлекает список узлов, которые содержат указанные метки.

5.3. `get_all_nodes(self)`

Назначение:
Получает все узлы без связей.

Запрос:

```
MATCH (a) RETURN a
```

5.4. `get_all_nodes_and_arcs(self)`

Назначение:
Возвращает все узлы с их исходящими связями.

Внутри вызывает: `_fetch_nodes_with_arcs`

5.5. `get_node_by_uri(self, uri)`

Назначение:

Находит узел по его `uri` и возвращает его со связями.

5.6. `update_node(self, uri, params_to_update)`

Назначение:
Обновляет свойства узла с заданным `uri`.

Cypher-запрос:

```
MATCH (n {uri: $uri})

SET n += $updates

RETURN n
```

5.7. `delete_node_by_uri(self, uri)`

Назначение:
Удаляет узел и все его связи по `uri`.

Cypher-запрос:

```
MATCH (n {uri: $uri})

DETACH DELETE n

RETURN count(n) as deleted_count
```

6. Операции со связями

6.1. `create_arc(self, node1_uri, node2_uri, rel_type)`

Назначение:
Создаёт направленную связь между двумя узлами по их URI.

Cypher-запрос:

```
MATCH (a {uri: $uri1}), (b {uri: $uri2})
```

```
CREATE (a)-[r:`TYPE`]->(b)
RETURN r, a, b
```

6.2. `delete_arc_by_id(self, arc_id)`

Назначение:
Удаляет связь по её идентификатору.

Cypher-запрос:

```
MATCH ()-[r]-() WHERE elementId(r) = $id
DELETE r
RETURN count(r) as deleted_count
```

7. Произвольные запросы

`run_custom_query(self, query, params=None)`

Назначение:
Позволяет выполнять любые Cypher-запросы к базе данных.

Пример использования:

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

8. Заключение

Класс **`Neo4jRepository`** реализует собственный драйвер для работы с графовой базой данных Neo4j