Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО Ульяновский государственный технический университет

Кафедра «Вычислительная техника»

Лабораторная работа №3

«Построение онтологической модели»

Вариант №24

Выполнил студент

группы ИВТИИбд-12

Яруллин Р.Х.

Преподаватель

Хайрулин И.Д.

Ульяновск, 2025

**§1. Структурное описание онтологической модели**

Онтологическая модель описывает строение автомобиля и формализует ключевые объекты и их взаимосвязи. Основные сущности (далее классы) онтологической модели:

* **Кузов («Body»):** виды кузовов
* **Двигатель («Engine»):** виды двигателей
* **Коробки передачи («Transmissions»):** виды коробок передач
* **Колеса («Wheels»):** виды колёс

Для реализации модели использованы два инструмента:

* **Protege** — применяется для построения онтологии, добавления аннотаций и формулировки логических правил с использованием SWRL.
* **Neo4j** — используется для создания графовой модели с узлами и связями, где логические зависимости реализуются с помощью языка Cypher.

**§2. Классы, отношения и аксиомы**

Онтологическая модель включает несколько типов элементов: классы, свойства данных и свойства объектов, а также логические правила, описывающие взаимосвязи между объектами.

§**2.1 Классы**

Модель содержит следующие основные классы:

* **Кузов («Body»):** виды кузовов
* **Двигатель («Engine»):** виды двигателей
* **Коробки передачи («Transmissions»):** виды коробок передач
* **Колеса («Wheels»):** виды колёс

Каждый класс является основной единицей модели и служит основой для определения свойств и отношений.

§**2.2 Свойства данных**

Свойства данных описывают характеристики отдельных объектов (индивидуумов) классов:

| *Свойство* | *К какому классу применимо* | *Описание* |
| --- | --- | --- |
| name | Body, Engine, Transmissions, Wheels | Наименование всех элементов авто |
| number\_of\_seats | Body | Количество посадочных мест |
| power\_in\_hp | Engine | Мощность двигателя |
| type | Transmissions, Body | Тип КПП, класса авто |
| type\_of\_road | Wheels | Тип дороги |

Эти свойства позволяют хранить конкретные данные для каждого объекта и использовать их при поиске и логическом выводе.

§**2.3 Свойства объектов**

Свойства объектов определяют связи между различными объектами модели:

| *Свойство* | *Описание* |
| --- | --- |
| IS\_TYPE\_OF | Является типом |
| IS\_INSTALLED\_IN | Установлен в |
| IS\_INSTALLED\_WITH | Установлен совместно с |
| TRANSMIT\_TORQUE\_TO | Передает крутящий момент на |

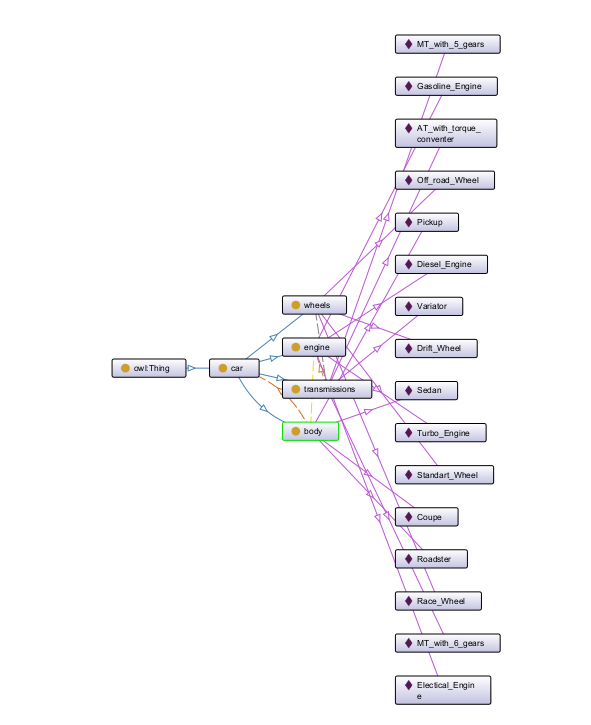
Эти свойства обеспечивают логическую структуру модели и позволяют выводить новые связи между объектами.

§**2.3 Реализация в Protégé**

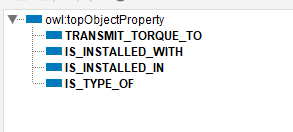
Для построения онтологической модели использовался инструмент **Protégé**, который позволяет создавать OWL-онтологии, задавать свойства и правила, а также выполнять логический вывод.

**Рисунок 1. Дерево классов и экземпляров**

Отображает иерархию и связи всех классов и экземпляров модели: Body, Engine, Transmission, Wheels.



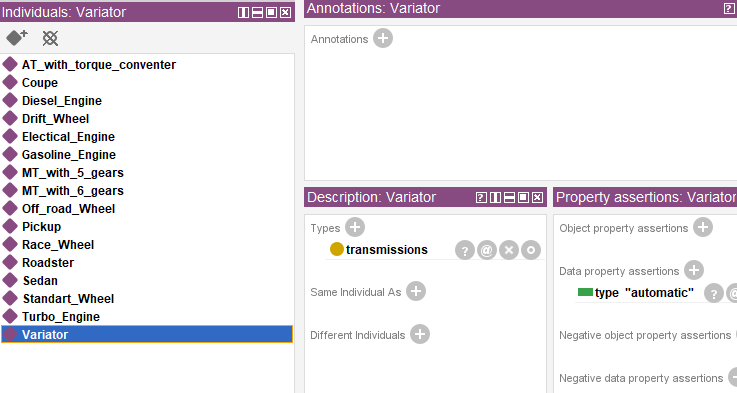
**Рисунок 2. Свойства объектов**

Показывает все свойства объектов.

**Рисунок 3. Экземпляры классов**

Отображает конкретные экземпляры классов:

* **Body:** Sedan, Coupe, Roadster, Pickup
* **Engine:** Gasoline\_Engine, Diesel\_Engine, Turbo\_Engine, Electrical\_Engine
* **Transmissions:** MT\_with\_6\_gears, MT\_with\_5\_gears, AT\_with\_torque\_conventer, Variator
* **Wheels:** Off\_road\_Wheels, Drift\_Wheels, Race\_Wheels, Standart\_Wheels



# §4. Реализация в Neo4j

Для построения графовой модели использовалась среда **Neo4j**, позволяющая хранить данные в виде узлов и связей.

**Создание узлов и связей:**

*CREATE*

*(car: Body {name: "Cars bodies", type: "not trucks"}),*

*(sedan: Body {name: "Sedan", number\_of\_seats: "Four"}),*

*(coupe: Body {name: "Coupe", number\_of\_seats: "Two"}),*

*(roadster: Body {name: "Roadster", number\_of\_seats: "Two"}),*

*(pickup: Body {name: "Pickup", number\_of\_seats: "Four"}),*

*(diesel\_engine: Engine {name: "Diesel engine", power\_in\_hp: "200"}),*

*(gasoline\_engine: Engine {name: "Gasoline engine", power\_in\_hp: "320"}),*

*(turbo\_engine: Engine {name: "Turbo engine", power\_in\_hp: "380"}),*

*(electic\_engine: Engine {name: "Electrical engine", power\_in\_hp: "120"}),*

*(offroad\_wheels: Wheels {name: "Off-road wheel", type\_of\_road: "off-road"}),*

*(standart\_wheels: Wheels {name: "Standart wheel", type\_of\_road: "all roads"}),*

*(race\_wheels: Wheels {name: "Race wheel", type\_of\_road: "race track"}),*

*(drift\_wheels: Wheels {name: "Drift wheel", type\_of\_road: "drift track"}),*

*(hydrotrans\_at: Transmissions {name: "AT with torque conventer", type: "automatic"}),*

*(var\_at: Transmissions {name: "Variator", type: "automatic"}),*

*(five\_mt: Transmissions {name: "MT with 5 gears", type: "mechanical"}),*

*(six\_mt: Transmissions {name: "MT with 6 gears", type: "mechanical"}),*

*(sedan) -[:IS\_TYPE\_OF]-> (car),*

*(coupe) -[:IS\_TYPE\_OF]-> (car),*

*(pickup) -[:IS\_TYPE\_OF]-> (car),*

*(roadster) -[:IS\_TYPE\_OF]-> (car),*

*(diesel\_engine) -[:IS\_INSTALLED\_IN]-> (pickup),*

*(gasoline\_engine) -[:IS\_INSTALLED\_IN]-> (sedan),*

*(turbo\_engine) -[:IS\_INSTALLED\_IN]-> (roadster),*

*(electic\_engine) -[:IS\_INSTALLED\_IN]-> (coupe),*

*(var\_at) -[:IS\_INSTALLED\_WITH]-> (gasoline\_engine),*

*(hydrotrans\_at) -[:IS\_INSTALLED\_WITH]-> (electic\_engine),*

*(six\_mt) -[:IS\_INSTALLED\_WITH]-> (diesel\_engine),*

*(five\_mt) -[:IS\_INSTALLED\_WITH]-> (turbo\_engine),*

*(var\_at) -[:TRANSMIT\_TORQUE\_TO]-> (drift\_wheels),*

*(hydrotrans\_at) -[:TRANSMIT\_TORQUE\_TO]-> (standart\_wheels),*

*(six\_mt) -[:TRANSMIT\_TORQUE\_TO]-> (offroad\_wheels),*

*(five\_mt) -[:TRANSMIT\_TORQUE\_TO]-> (race\_wheels)*

**Рисунок 5. Демонстрация связей**

Визуализация графа показывает все узлы и их связи.



**§5. Вывод**

В работе создана онтологическая модель, объединяющая элементы автомобиля

Модель наглядно реализована в **Protege** и **Neo4j**, что подтверждается визуализацией и скриншотами.

Модель помогает рассмотреть взаимодействие между различными элементами.