

SWRL – это язык правил, который позволяет формулировать логические правила на основе OWL (Web Ontology Language) и RDF (Resource Description Framework).

SWRL используется для вывода новых знаний на основе существующих данных в онтологиях

### **Основные компоненты SWRL**

#### **1. Классы и экземпляры:**

Классы обозначаются с помощью `ClassName(?x)`, где `?x` — это переменная, представляющая экземпляр класса.

*Пример:* `Person(?p)` означает, что `?p` является экземпляром класса `Person`.

#### **2. Свойства (параметры):**

Свойства используются для связывания экземпляров. Они могут быть объектными или свойствами данных.

*Пример:* `hasAge(?p, ?age)` означает, что у экземпляра `?p` есть свойство `hasAge`, значение которого хранится в переменной `?age`.

#### **3. Функции и предикаты:**

Для выполнения логических операций и сравнений используются встроенные функции, такие как `swrlb:greaterThan`, `swrlb:lessThan`, `swrlb:equal`, и т.д.

*Пример:* `swrlb:greaterThan(?age, 18)` проверяет, больше ли значение переменной `?age` 18.

#### **4. Логические операторы:**

Правила могут содержать несколько условий, соединённых логическими операторами:

$\wedge$  (и): используется для объединения нескольких условий.

$\vee$  (или): используется для обозначения альтернативных условий.

*Пример:* `Person(?p)  $\wedge$  hasAge(?p, ?age)  $\wedge$  swrlb:greaterThan(?age, 18)` означает, что все три условия должны быть истинными.

## 5. Выводы (заключения):

Заключение записывается после символа  $\rightarrow$ . Оно указывает, что если все условия истинны, то вывод также будет истинным.

*Пример:*  $\rightarrow \text{Adult}(?p)$  означает, что если все условия слева от стрелки истинны, то  $?p$  будет также являться экземпляром класса `Adult`.

### Структура правил SWRL:

Правила имеют следующую структуру:  $\text{Body} \rightarrow \text{Head}$ ,

где `Body` – это условия, которые должны быть выполнены, `Head` – это вывод, который следует сделать, если условия выполнены.

*Пример правила:*

$\text{Person}(?p) \wedge \text{hasAge}(?p, ?a) \wedge \text{swrlb:greaterThan}(?a, 18) \rightarrow \text{Adult}(?p)$

Это правило говорит о том, что если существует человек с возрастом больше 18 лет, то он является взрослым.

*Пример правила:*

$\text{Person}(?p) \wedge \text{hasAge}(?p, ?age) \wedge (\text{swrlb:lessThan}(?age, 18) \vee \text{swrlb:greaterThan}(?age, 65)) \rightarrow \text{YouthOrSenior}(?p)$

Это правило говорит о том, что если существует человек с возрастом меньше 18 или больше 65, то он относится к категории “`YouthOrSenior`”.

### Встроенные предикаты SWRL

#### 1. *swrlb:equal*

Описание: Проверяет, равны ли два значения.

Пример:  $\text{swrlb:equal}(?a, 10)$  – возвращает `true`, если  $?a$  равно 10.

#### 2. *swrlb:notEqual*

Описание: Проверяет, не равны ли два значения.

Пример:  $\text{swrlb:notEqual}(?a, 10)$  – возвращает `true`, если  $?a$  не равно 10.

#### 3. *swrlb:greaterThan*

Описание: Проверяет, больше ли одно значение другого.

Пример:  $\text{swrlb:greaterThan}(?a, ?b)$  – возвращает `true`, если  $?a$  больше  $?b$ .

#### *4. swrlb:greaterThanOrEqual*

Описание: Проверяет, больше ли одно значение или равно другому.

Пример: `swrlb:greaterThanOrEqual(?a, 10)` – возвращает `true`, если `?a` больше или равно 10.

#### *5. swrlb:lessThan*

Описание: Проверяет, меньше ли одно значение другого.

Пример: `swrlb:lessThan(?a, ?b)` – возвращает `true`, если `?a` меньше `?b`.

#### *6. swrlb:lessThanOrEqual*

Описание: Проверяет, меньше ли одно значение или равно другому.

Пример: `swrlb:lessThanOrEqual(?a, 10)` – возвращает `true`, если `?a` меньше или равно 10.

#### *7. swrlb:add*

Описание: Суммирует два числовых значения.

Пример: `swrlb:add(?a, ?b, ?c)` – где `?c` будет равен сумме `?a` и `?b`.

#### *8. swrlb:subtract*

Описание: Вычитает одно числовое значение из другого.

Пример использования: `swrlb:subtract(?a, ?b, ?c)` – где `?c` будет равен разности `?a` и `?b`.

#### *9. swrlb:multiply*

Описание: Умножает два числовых значения.

Пример: `swrlb:multiply(?a, ?b, ?c)` – где `?c` будет равен произведению `?a` и `?b`.

#### *10. swrlb:divide*

Описание: Делит одно числовое значение на другое.

Пример использования: `swrlb:divide(?a, ?b, ?c)` – где `?c` будет равен частному `?a` и `?b`.

#### *11. swrlb:stringLength*

Описание: Возвращает длину строки.

Пример: `swrlb:stringLength(?str, ?len)` – где `?len` будет длиной строки `?str`.

### 12. *swrlb:substring*

Описание: Извлекает подстроку из строки.

Пример: *swrlb:substring*(?str, ?start, ?length, ?subStr) – где ?subStr будет подстрокой длиной ?length, начиная с позиции ?start в строке ?str.

### 13. *swrlb:toUpperCase*

Описание: Преобразует строку в верхний регистр.

Пример: *swrlb:toUpperCase*(?str, ?upperStr) – где ?upperStr будет строкой в верхнем регистре.

### 14. *swrlb:toLowerCase*

Описание: Преобразует строку в нижний регистр.

Пример: *swrlb:toLowerCase*(?str, ?lowerStr) – где ?lowerStr будет строкой в нижнем регистре.

### 15. *swrlb:contains*

Описание: Проверяет, содержит ли первый аргумент второй аргумент (с учётом регистра). Также существует функция *swrlb:containsIgnoreCase*, которая проверяет, содержит ли первый аргумент второй аргумент (с игнорированием регистра)

Пример: *swrlb:contains*(?meda, "Плановый осмотр")

## Примеры:

Создадим класс **Мужчина** и определим все индивидуалы, которые имеют значение предиката «Пол» = «Мужской» в этот класс.

Пациент(?pat) ^ Пол(?pat, "Мужской") -> Мужчина(?pat)

## ЛекарстваВысокойЭффективности:

Результаты\_клинических\_испытаний(?medrez) ^

Исследовалось(?medrez, ?medlek) ^ ПоказательБезопасности(?medrez, ?medbez) ^ ПоказательЭффективности(?medrez, ?medeff) ^

*swrlb:greaterThanOrEqualTo*(?medbez, 0.9) ^ *swrlb:greaterThanOrEqualTo*(?medeff, 0.9) -> ЛекарстваВысокойЭффективности(?medlek)

**Если у пациента диагноз «Пневмония», то он «ТребуетсяИзоляции».**

Медицинская\_карта\_запись (?mk) ^ принадлежитПациенту(?mk,?pat) ^  
основноеЗаболеваниеПриОбращении (?mk, ?zab) ^  
НазваниеЗаболевания(?zab, "Пневмония") -> ТребуетсяИзоляции(?pat)

**Если у пациента Плановый осмотр, то ему нужно  
«НаправлениеНаПлановыйОсмотр»**

Медицинская\_карта\_запись(?medmk) ^ принадлежитПациенту(?medmk,  
?medpat) ^ ОсновнаяЖалоба(?medmk, ?meda) ^ swrlb:contains(?meda,  
"Плановый осмотр") -> НаправлениеНаПлановыйОсмотр(?medpat)

**Если препарат используется для лечения пневмонии, то он должен  
быть протестирован на взрослых.**

Показания\_к\_применению (?pp) ^ ПрименениеЛекарства (?pp, ?lek) ^  
ДляЗаболевания(?pp, ?zab) ^ НазваниеЗаболевания(?zab,?nazab) ^  
swrlb:contains(?nazab, "Пневмония") -> ПротестированНаВзрослых (?lek)

**Если препарат имеет побочный эффект «головокружение», то он  
должен быть помечен как требующий мониторинга**

Результаты\_клинических\_испытаний(?medrez) ^ Исследовалось(?medrez,  
?medlek) ^ ОбщиеПобочныеЭффекты(?medrez, "Головокружение") ->  
ТребуетсяМониторинга(?medlek, true)

## **Задания**

1. Если препарат имеет эффективность ниже 0.7, то он не  
рекомендуется.

2. Если препарат имеет побочные эффекты и эффективность выше 0.8, то его следует использовать с осторожностью.

3. Если препарат имеет эффективность ниже 0.6 и безопасность ниже 0.7, то его следует исключить из списка препаратов.

4. Создать список пациентов по группе крови.

5. Если женщина имеет статус «Беременная», то она «ПодОсобымНаблюдением».

6. Если пациент имеет более трех симптомов то он "ПациентЭкстреннойСитуации".

7. Если препарат не исследовался, то он должен быть помечен как требующий исследований.