

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Филологический факультет

УТВЕРЖДЕНО:

Декан

И. В. Тубалова

Оценочные материалы по дисциплине

Формальная семантика. Онтологии и тезаурусы

по направлению подготовки

45.04.03 Фундаментальная и прикладная лингвистика

Направленность (профиль) подготовки:

Компьютерная и когнитивная лингвистика

Форма обучения

Очная

Квалификация

Магистр

Год приема

2025

СОГЛАСОВАНО:

Руководитель ОП

З.И. Резанова

Председатель УМК

Ю.А. Тихомирова

Томск – 2025

1. Компетенции и индикаторы их достижения, проверяемые данными оценочными материалами

Целью освоения дисциплины является формирование следующих компетенций:

ПК-3 Способен разрабатывать системы автоматической обработки звучащей речи и письменного текста на естественном языке, лингвистические компоненты электронных ресурсов и интеллектуальных электронных систем (лингвистические корпуса, словари, онтологии, базы данных).

ПК-4 Способен разрабатывать проекты прикладной направленности в области когнитивной и компьютерной лингвистики с применением современных технических средств и информационных технологий, в том числе в области искусственного интеллекта.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИПК-3.3 Разрабатывает лингвистические компоненты интеллектуальных информационных систем (онтологии, базы данных)

ИПК-4.2 Разрабатывает программу действий по решению задач проекта в области когнитивной и компьютерной лингвистики с учетом имеющихся технических средств и информационных технологий, в том числе в области искусственного интеллекта

ИПК-4.3 Обеспечивает выполнение проекта в области когнитивной и компьютерной лингвистики с применением современных технических средств и информационных технологий, в том числе в области искусственного интеллекта, в соответствии с установленными целями, сроками и затратами

2. Оценочные материалы текущего контроля и критерии оценивания

Элементы текущего контроля:

- тесты;
- контрольная работа;

Тест (ИПК-3.3.)

1. Что такое онтология в контексте инженерии знаний?
 - а) Система управления базами данных
 - б) Язык программирования
 - в) Формальное представление знаний
 - г) графический интерфейс пользователя
2. Какой язык обычно используется для представления онтологий?
 - а) HTML
 - б) XML
 - в) Java
 - г) Python
3. Какова цель разработки онтологий?
 - а) Для разработки пользовательских интерфейсов
 - б) Для создания веб-приложений
 - с) Для создания моделей знаний
 - г) Для оптимизации производительности баз данных
4. Что из перечисленного ниже является ключевым компонентом онтологии?
 - а) Экземпляры
 - б) Функции
 - в) Циклы
 - г) Переменные

5. Как называется процесс определения отношений между понятиями в онтологии?

- а) Вывод
- б) Классификация
- в) рассуждение
- г) Таксономия

Ключи:

- 1. в
- 2. б
- 3. в
- 4. а
- 5. г

Критерии оценивания: тест считается пройденным, если обучающий ответил правильно как минимум на половину вопросов.

Контрольная работа (ИПК-4.2, ИПК-4.3)

Контрольная работа состоит из 2 задач.

Примеры задач:

Задача 1 (ИПК-4.2)

Опишите процесс создания трех классов в редакторе Protege 5.

Ответ

- 1. Перейдите к вкладке Сущности.
- 2. Нажмите кнопку +. Эта кнопка создает новый подкласс выбранного подкласса. В данном случае мы хотим создать подкласс класса Вещь. Добавить подкласс.
- 3. Должно появиться диалоговое окно. Затем выберите Создать новый класс с полем для имени нового класса.
- 4. Повторите предыдущие шаги, чтобы добавить еще два класса.

Задача 2 (ИПК-4.3)

Напишите заголовок rdf-файла с декларацией необходимых пространств имен для соответствия стандартам W3C.

Ответ

```
xml:base="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/newpizza/"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
```

Критерии оценивания:

Результаты контрольной работы определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

При оценивании ответа главное внимание уделяется демонстрации обучающимся знаний о написании программного кода, специфики программы Protege и синтаксиса языка OWL2.

За каждый вопрос и практическое задание выставляется отдельная оценка, общая экзаменационная оценка складывается из трех частных оценок. Если студент получает оценку «неудовлетворительно» за один из вопросов или за практическое задание, общая положительная оценка не выставляется.

Оценка	Критерии	
	Теоретический вопрос	Практическое задание
Отлично	Ответ полный, проиллюстрированный самостоятельно подобранными примерами, содержание структурировано, логика ответа прозрачна.	Задание выполнено без ошибок или допущена 1 незначительная ошибка
Хорошо	А) Ответ полный, но не структурированный, примеры заимствованы из разобранных на занятиях или некачественно проинтерпретированы. Б) В теоретической части ответа имеются отдельные лакуны, которые могут быть заполнены на основании дополнительных вопросов.	Задание выполнено с 2 ошибками, которые студент способен исправить после указания на нее.
Удовлетворительно	Ответ не полный, слабо структурированный, студент некачественно устанавливает связи обсуждаемой проблемы с другими положениями изученного курса, допускает грубые ошибки в интерпретации примеров.	Задание выполнено с 3 ошибками.
Неудовлетворительно	При ответе допускаются грубые теоретические ошибки, обнаруживаются пробелы в знаниях важнейших теоретических положений курса, студент не способен приводить и интерпретировать примеры.	Задание не выполнено либо при его выполнении допущено более 4 ошибок.

3. Оценочные материалы итогового контроля (промежуточной аттестации) и критерии оценивания

Третий семестр, экзамен

Экзаменационный билет состоит из двух частей.

Первая часть представляет собой батарею тестов из вопросов, проверяющих набор компетенций. Ответы на вопросы первой части даются путем выбора из списка предложенных.

Тест (ИПК-3.3)

1. Какой из перечисленных ниже языков является примером языка онтологий?

- а) OWL
- б) SQL
- в) C++
- г) Ruby

2. Какова цель выравнивания онтологий?

- а) Объединить несколько онтологий в одну
- б) Устранение несоответствий в онтологии
- в) Для оценки производительности онтологии
- г) Для графической визуализации онтологии

3. Что из перечисленного ниже является причиной использования онтологий в информационном поиске?

- а) Для улучшения дизайна веб-сайта
- б) Для улучшения оптимизации поисковых систем
- в) Для обеспечения семантического поиска
- г) Для увеличения пропускной способности сети

4. Как называется процесс создания онтологии из существующих источников данных?

- а) Извлечение онтологии
- б) обоснование онтологии
- в) Визуализация онтологии
- г) отображение онтологии

5. Какова цель оценки онтологии?

- а) Измерить размер онтологии
- б) Оценить качество онтологии
- в) Вычислить время выполнения онтологии
- г) Генерировать образцы данных для онтологии

Ответ

- 1. а
- 2. а
- 3. в
- 4. а
- 5. б

Вторая часть содержит задачи на программирование, проверяющее ИПК-4.2. и ИПК-4.3

Задача 1 (ИПК-4.2)

Напишите декларацию произвольного класса, входящего в произвольный подкласс.

Ответ:

```
<owl:Class
rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#OliveTopping">
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#VegetableTopping"/>
</owl:Class>
```

Задача 2 (ИПК-4.3)

Напишите декларацию произвольного анонимного класса, использующую ограничение на одно свойство.

Ответ

```
<!--
http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/newpizza#SlightlySpicyPizza --
>
<owl:Class
rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#SlightlySpicyPizza">
```

```

<owl:equivalentClass>
  <owl:Class>
    <owl:intersectionOf rdf:parseType="Collection">
      <rdf:Description
rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/newpizza#Pizza"/>
        <owl:Restriction>
          <owl:onProperty
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#hasTopping"/>
            <owl:someValuesFrom>
              <owl:Restriction>
                <owl:onProperty
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#hasSpiciness"/>
                  <owl:hasValue
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#Medium"/>
                </owl:Restriction>
              </owl:someValuesFrom>
            </owl:Restriction>
          </owl:intersectionOf>
        </owl:Class>
      </owl:equivalentClass>
    </owl:Class>

```

Критерии оценивания:

Результаты экзамена определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

При оценивании ответа на зачете с оценкой главное внимание уделяется демонстрации обучающимся знаний об особенностях работы с программой Protege и написании кода декларативного программирования в области формальной семантики.

За каждый вопрос и практическое задание выставляется отдельная оценка, общая экзаменационная оценка складывается из трех частных оценок. Если студент получает оценку «неудовлетворительно» за один из вопросов или за практическое задание, общая положительная оценка не выставляется.

Оценка	Критерии	
	Теоретический вопрос	Практическое задание
Отлично	Ответ полный, проиллюстрированный самостоятельно подобранными примерами, содержание структурировано, логика ответа прозрачна.	Задание выполнено без ошибок или допущена 1 незначительная ошибка
Хорошо	А) Ответ полный, но не структурированный, примеры заимствованы из разобранных на занятиях или некачественно проинтерпретированы. Б) В теоретической части ответа имеются отдельные лакуны, которые могут быть заполнены на основании дополнительных вопросов.	Задание выполнено с 2 ошибками, которые студент способен исправить после указания на нее.
Удовлетворитель но	Ответ не полный, слабо структурированный, студент	Задание выполнено с 3 ошибками.

	некачественно устанавливает связи обсуждаемой проблемы с другими положениями изученного курса, допускает грубые ошибки в интерпретации примеров.	
Неудовлетворительно	При ответе допускаются грубые теоретические ошибки, обнаруживаются пробелы в знаниях важнейших теоретических положений курса, студент не способен приводить и интерпретировать примеры.	Задание не выполнено либо при его выполнении допущено более 4 ошибок.

4. Оценочные материалы для проверки остаточных знаний (сформированности компетенций)

5 заданий, проверяющих ИПК-3.3 Разрабатывает лингвистические компоненты интеллектуальных информационных систем (онтологии, базы данных).

Задание 1

1. Что такое онтология в контексте инженерии знаний?
 - а) Система управления базами данных
 - б) Язык программирования
 - в) Формальное представление знаний
 - г) графический интерфейс пользователя
2. Какой язык обычно используется для представления онтологий?
 - а) HTML
 - б) XML
 - в) Java
 - г) Python
3. Какова цель разработки онтологий?
 - а) Для разработки пользовательских интерфейсов
 - б) Для создания веб-приложений
 - с) Для создания моделей знаний
 - г) Для оптимизации производительности баз данных

Ответ

1. в
2. б
3. в

Задание 2

1. Что из перечисленного ниже является ключевым компонентом онтологии?
 - а) Экземпляры
 - б) Функции
 - в) Циклы
 - г) Переменные
2. Как называется процесс определения отношений между понятиями в онтологии?
 - а) Вывод

- б) Классификация
- в) рассуждение
- г) Таксономия

3. Какой из перечисленных ниже языков является примером языка онтологий?

- а) OWL
- б) SQL
- с) C++
- г) Ruby

Ответ

- 1. а
- 2. г
- 3. а

Задание 3

1. Какова цель выравнивания онтологий?

- а) Объединить несколько онтологий в одну
- б) Устранение несоответствий в онтологии
- в) Для оценки производительности онтологии
- г) Для графической визуализации онтологии

2. Что из перечисленного ниже является причиной использования онтологий в информационном поиске?

- а) Для улучшения дизайна веб-сайта
- б) Для улучшения оптимизации поисковых систем
- в) Для обеспечения семантического поиска
- г) Для увеличения пропускной способности сети

3. Как называется процесс создания онтологии из существующих источников данных?

- а) Извлечение онтологии
- б) обоснование онтологии
- в) Визуализация онтологии
- г) отображение онтологии

4. Какова цель оценки онтологии?

- а) Измерить размер онтологии
- б) Оценить качество онтологии
- в) Вычислить время выполнения онтологии
- г) Генерировать образцы данных для онтологии

Ответ

- 1. а
- 2. в
- 3. а
- 4. б

Задание 4

1. В контексте баз данных каково назначение онтологии?
 - а) Для определения структуры и взаимосвязей данных
 - б) Оптимизировать производительность запросов
 - с) Обеспечение ограничений целостности данных
 - д) Предоставить удобный интерфейс для работы с данными.
2. Какой из следующих языков обычно используется для запросов к онтологиям?
 - а) SPARQL
 - б) SQL
 - в) Java
 - г) Python
3. Как называется процесс отображения онтологий на реляционные базы данных?
 - а) Извлечение онтологии
 - б) Семантическое отображение
 - в) Нормализация данных
 - г) эволюция схем
4. Что из перечисленного ниже НЕ является типом отображения онтологии?
 - а) отображение экземпляров
 - б) отображение классов
 - в) отображение свойств
 - г) отображение индексов

Ответ

1. а
2. а
3. б
4. г

Задание 5

1. Какова цель интеграции данных на основе онтологий?
 - а) Объединить данные из различных источников в единое представление
 - б) Оптимизировать производительность базы данных
 - в) Обеспечить соблюдение ограничений ссылочной целостности
 - г) Генерировать образцы данных для тестирования
2. Какой из перечисленных ниже языков является примером языка онтологий, используемого для баз данных?
 - а) OWL
 - б) XML
 - в) JSON
 - г) RDF
3. В чем основное отличие баз данных на основе онтологий от традиционных реляционных баз данных?
 - а) Базы данных на основе онтологий хранят данные в виде графовой структуры
 - б) Традиционные реляционные базы данных имеют более высокую производительность
 - в) Базы данных на основе онтологий не поддерживают транзакции
 - г) Традиционные реляционные базы данных более масштабируемы

4. Как называется процесс рассуждений с использованием онтологий для получения новых знаний?

- а) Выравнивание онтологий
- б) извлечение онтологий
- в) рассуждение на основе онтологий
- г) Визуализация онтологий

Ответ:

- 1. а
- 2. а
- 3. а
- 4. в

5 заданий, проверяющих ИПК-4.2 Разрабатывает программу действий по решению задач проекта в области когнитивной и компьютерной лингвистики с учетом имеющихся технических средств и информационных технологий, в том числе в области искусственного интеллекта.

Задание 1

Определите домен и диапазон свойства hasTopping.

Ответ

- 1. Перейдите к Свойства объекта вкладки. Выберите свойство hasTopping. \
- 2. Нажмите значок «Добавить». Будет представлено окно с несколькими вкладками. Существует несколько способов определения домена и диапазона. Сейчас мы воспользуемся самым простым методом (и наиболее часто используемым). Выберите вкладку. Теперь вы должны увидеть (+) рядом с Домены (пересечение). Нажмите на Пицца под Домены в Описание.
- 3. Повторите шаг 2, но на этот раз начните с hasTopping. На этот раз выберите класс (+) значок рядом с Диапазоны (пересечение) Описание ПиццаТоппинг как диапазон.
- 4. Синхронизируйте ризонер. Вы должны увидеть, что Домен и Диапазон для isToppingОфибыли заполнены ризонером. Поскольку эти два свойства являются обратными, рассуждающий знает, что областью определения одного из них является диапазон другого, и наоборот. Это еще один пример того, почему частое выполнение рассуждения может сэкономить время и помочь поддерживать корректность модели.

Задание 2

Используйте инструмент «Создать иерархию классов» для создания подклассов PizzaBase.

Ответ

- 1. Выберите класс
- 2. С PizzaBase в иерархии классов в качестве выбранного, используйте Инструменты> опция меню Создать иерархию классов.
- 3. Должен появиться мастер, который позволит вам одновременно создать вложенную группу классов. Вы должны увидеть окно с надписью где вы можете ввести одно имя в каждой строке. Вы также можете использовать клавишу табуляции, чтобы указать, что класс является подклассом класса, расположенного над ним. Сейчас мы просто хотим ввести два подкласса PizzaBase: ThinAndCrispyBase и DeepPanBase. Одна из задач мастера — это автоматическое добавление для нас префикса или суффикса. В поле «Суффикс» добавьте Base.
- 4. Значение по умолчанию должно быть отмечено как непересекающееся, что нам и нужно в данном случае (основа не может быть одновременно глубокой и тонкой). Синхронизируйте рассуждение.

Задание 3

Напишите фрагмент декларативного кода для индивида, принадлежащего одному подклассу.

Ответ

```
<!--  
http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/newpizza#AmericanaHotPizza1  
-->  
  
                                <owl:NamedIndividual  
rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/  
newpizza#AmericanaHotPizza1">  
    <rdf:type    rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/  
2023/10/newpizza#AmericanaHotPizza"/>  
                                <newpizza:hasCaloricContent  
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">563</newpizza:hasCaloricCont  
ent>  
                                </owl:NamedIndividual>
```

Задание 4

Напишите декларативный код произвольного класса, принадлежащего произвольному подклассу.

Ответ

```
                                <owl:Class  
rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/  
newpizza#TunaTopping">  
                                <rdfs:subClassOf  
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/  
newpizza#SeafoodTopping"/>  
                                </owl:Class>
```

Задание 5

1. Что из нижеперечисленного является примером языка запросов на основе онтологий?

- а) SPARQL
- б) SQL
- в) Python
- г) C#

2. Какова цель оценки онтологии в контексте баз данных?

- а) Измерить сложность онтологии
- б) Оценить качество и корректность онтологии
- в) Для оптимизации индексирования баз данных
- г) Генерировать синтетические данные для тестирования

1. а

2. б

5 заданий, проверяющих ИПК-4.3 Обеспечивает выполнение проекта в области когнитивной и компьютерной лингвистики с применением современных технических средств и информационных технологий, в том числе в области искусственного интеллекта, в соответствии с установленными целями, сроками и затратами.

Задание 1

Добавьте ограничение к Pizza, указывающее, что Pizza должна иметь PizzaBase.

Ответ

1. Выберите иерархию классов.
2. Нажмите значок «Добавить» рядом с вкладкой Классы.
3. Откроется новое окно с несколькими вкладками для определения нового ограничения. На этой вкладке есть создатель ограничения.
4. Разверните иерархию свойств слева и выберите Заполнитель ограничений справа выберите класс, который будет установлен. Это должно быть значение по умолчанию, поэтому вам не нужно ничего менять. hasBase как свойство, подлежащее ограничению. Наконец, тип ограничения внизу должен быть some (экзистенциальные)
5. Нажмите кнопку. Это должно закрыть окно. Запустите программу рассуждения, чтобы убедиться, что все согласовано.

Задание 2

Добавьте непересекающиеся классы Пицца, ПиццаТоппинг и ПиццаБаза к онтологии.

Ответ

1. Выберите класс Пицца в иерархии классов.
2. Найдите вариант в кружок Непересекающийся с Описание просмотреть и выбрать (+) подпишите рядом с ним. Смотри на красный.
3. Должно открыться диалоговое окно с двумя вкладками: на данный момент (редактор выражений мы воспользуемся позже). Это дает вам интерфейс для выбора класса это идентично представлению иерархии классов. Используйте его для навигации и выбора Иерархия классов и Редактор выражений . Вы хотите Сорт иерархия ПиццаБаза. Удерживайте клавишу Shift и ПиццаБаза и ПиццаТоппинг.
4. Сделайте что каждый из них имеет соответствующие непересекающиеся аксиомы, определенные для указания того, что каждый из этих классов не пересекается с двумя другими. Reasoner>Синхронизировать Reasoner

Задание 3

Установите и запустите Pellet Reasoner

Ответ

1. Проверить, установлен ли генератор Pellet. Нажмите на значок, появится список установленных ризонеров, таких как меню, затем выбрать его.
2. Меню ризонеров. В нижней части меню будет Hermit и возможно Pellet. Если Pellet виден в этом
2. Если Pellet не отображается, установить Pellet через меню плагинов, и получить сообщение о том, что он вступит в силу в следующий раз, когда при запуске Protégé. Сохранить работу, закрыть Protégé и перезапустить его. Затем перейдите к меню плагинов-ризонеров. Увидеть сохраненную онтологию в списке последних онтологий. Загрузить ее. Теперь мы должны увидеть Pellet под меню и иметь возможность выбрать его, сделать это.
3. Выбрав Pellet в меню Reasoner, выполнить команду Reasoner должен работать очень быстро, поскольку онтология очень простая. Вы заметите, что небольшое текстовое сообщение в правом нижнем углу окна Protégé изменилось и теперь говорит о том, что Reasoner активен. В следующий раз, когда вы внесете изменения в онтологию, текст изменится на следующее: Состояние Reasoner не синхронизировано. При использовании небольших онтологий процесс рассуждения работает очень быстро, поэтому неплохо было бы усвоить привычку запускать его часто, а также после каждого изменения.

4. Вполне возможно, что один или несколько ваших классов по-прежнему будут выделены красным после запуска генератора рассуждений. Если это произойдет, сделайте Окно>Обновить пользовательский интерфейс и любые красные блики должны исчезнуть.

5. И последнее, что нам нужно сделать, это настроить генератор рассуждений. По умолчанию модуль рассуждения не выполняет все возможные выводы, поскольку некоторые выводы могут занять много времени для больших и сложных онтологий.

Задание 4

Напишите декларативный код, объявляющий класс Вегетарианской пиццы, как любой пиццы, не содержащей мяса.

Ответ

```
<owl:Class rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/newpizza#VegetarianPizza">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <rdf:Description
          rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/newpizza#Pizza"/>
          <owl:Restriction>
            <owl:onProperty
              rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/newpizza#hasTopping"/>
            <owl:allValuesFrom>
              <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                  <rdf:Description
                    rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/newpizza#CheeseTopping"/>
                  <rdf:Description
                    rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/newpizza#VegetableTopping"/>
                </owl:unionOf>
              </owl:Class>
            </owl:allValuesFrom>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
```

Задание 5

Создайте класс конкретных пицц и наполните его.

Ответ

```
<owl:Class rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/newpizza#SohoPizza">
  <rdfs:subClassOf
    rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/newpizza#NamedPizza"/>
  <rdfs:subClassOf>
    <owl:Restriction>
```

```

                                                                    <owl:onProperty
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#hasTopping"/>

                                                                    <owl:someValuesFrom
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#MozzarellaTopping"/>
    </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
    <owl:Restriction>

                                                                    <owl:onProperty
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#hasTopping"/>

                                                                    <owl:someValuesFrom
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#OliveTopping"/>
    </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
    <owl:Restriction>

                                                                    <owl:onProperty
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#hasTopping"/>

                                                                    <owl:someValuesFrom
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#ParmezanTopping"/>
    </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
    <owl:Restriction>

                                                                    <owl:onProperty
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#hasTopping"/>

                                                                    <owl:someValuesFrom
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#TomatoTopping"/>
    </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
    <owl:Restriction>
    <owl:onProperty
rdf:resource="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#hasTopping"/>
    <owl:allValuesFrom>
    <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
    <rdf:Description
rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#MozzarellaTopping"/>
    <rdf:Description
rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#OliveTopping"/>

```

```

        <rdf:Description
rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#ParmezanTopping"/>
        <rdf:Description
rdf:about="http://www.semanticweb.org/dyats_ui8a8rg/ontologies/2023/10/
newpizza#TomatoTopping"/>
        </owl:unionOf>
        </owl:Class>
        </owl:allValuesFrom>
        </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:comment>A pizza that only has Mozzarella, Tomato, Pepperoni, Olive and
Parmesan toppings.</rdfs:comment>
        </owl:Class>

```

Информация о разработчиках

Шиляев Константин Сергеевич, к. филол. н., доцент, кафедра общей, компьютерной и когнитивной лингвистики, инвалид умственного труда