

Домашно бр. 3
OWL

А) Домашна задача

1. Што е онтологија, а што јазик за опис на онтологија?

Онтологија: Онтологијата е формален опис на знаењето за одредена доменска област, кој вклучува концепти (класи), нивните својства (атрибути), релации меѓу концептите и ограничувања што се применуваат на нив. Онтологиите се користат за структурирање и претставување на знаење на начин што може да се разбере од машини.

Јазик за опис на онтологија: Јазик за опис на онтологија е формален јазик кој се користи за специфицирање и претставување на онтологии. Таквиот јазик обезбедува синтакса и семантика за дефинирање класи, релации и ограничувања. Примери за такви јазици се RDF, RDFS, и OWL.

2. Нацртајте го односот / компатибилноста на OWL класи и својства со RDF/RDFS.

OWL е надградба врз основа на RDF и RDFS и ги проширува нивните можности:

- **RDF:** Основна структура за репрезентација на податоци како трплети (субјект-предикат-објект).
- **RDFS:** Додава можности за дефинирање класи, хиерархии на класи (`rdfs:subClassOf`), и хиерархии на својства (`rdfs:subPropertyOf`).
- **OWL:** Додава поголема експресивност преку дефинирање сложени односи, ограничувања (на пр., кардиналност), логички изрази (`union`, `intersection`), и инверзни релации.



3. Наведете ги различните типови својства кои постојат во OWL.

OWL поддржува различни типови својства:

1. **Object Properties:** Релации помеѓу две класи (на пр., `hasFriend`).
2. **Datatype Properties:** Релации меѓу класа и литерална вредност (на пр., `hasAge`).
3. **Annotation Properties:** Користат за додавање метаподатоци на класи, својства или онтологијата (на пр., `rdfs:label`).
4. **Symmetric Properties:** Својства каде што ако А е поврзано со В, тогаш и В е поврзано со А.
5. **Transitive Properties:** Својства каде што ако А е поврзано со В, и В со С, тогаш А е поврзано со С.
6. **Functional Properties:** Својства кои можат да имаат најмногу една вредност за даден објект.
7. **Inverse Properties:** Релации кои се инверзни една на друга (на пр., `hasChild` и `isChildOf`).
8. **Irreflexive Properties:** Релации каде што објектот не може да се поврзе самиот со себе.

4. Напишете примери за дефинирање класа и релација во OWL. Нека класата има суперкласа, а релацијата нека биде инверзна на друга релација. Користете Turtle синтакса.

```
@prefix : <http://example.org/owl#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

:Person a owl:Class .
:Student a owl:Class ;
rdfs:subClassOf :Person .

:hasParent a owl:ObjectProperty ;
owl:inverseOf :hasChild .
:hasChild a owl:ObjectProperty .
```

5. Како се дефинираат ограничувања кај класите во OWL? Напишете еден пример за дефинирање ограничувања кај една OWL класа.

Ограничувањата во OWL се користат за дефинирање дополнителни услови за инстанци на класи. Примери:

- **Кардиналност:** Ограничува колку релации може да има објект.
- **Value Restriction:** Дефинира ограничувања за вредностите кои може да ги има релацијата.

```
@prefix : <http://example.org/owl#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

:Adult a owl:Class ;
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty :hasAge ;
    owl:allValuesFrom xsd:integer ;
    owl:minInclusive "18"^^xsd:integer
  ] .
```

Б) Практична задача

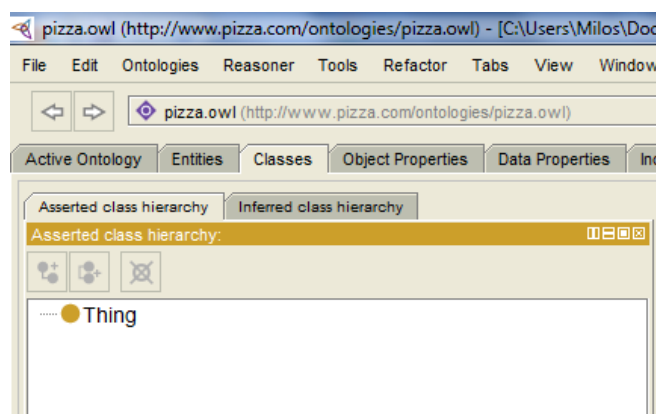
Вовед

[Protégé](#) е бесплатна, open-source платформа за креирање модели на домени и апликации со бази на знаење, базирани на онтологии. Protégé нуди поддршка за развој на секаков вид онтологии: колекции од хиерархиски поврзани термини, класификации, шеми на бази на податоци, итн.

1. Стартувајте го Protégé. Креирајте нова OWL онтологија:
 - a. Ontology URI: <http://www.pizza.com/ontologies/pizza.owl>
 - b. Локација: MyDocuments/WBS/pizza.owl

I. Дефинирање на класите во онтологијата

2. Отворете го табот Classes. Празното дрво на класи содржи само една класа, наречена owl:Thing, која е суперкласа на сите класи (Слика 1). Креирајте подкласи Pizza, PizzaTopping и PizzaBase. Сите се подкласи на owl:Thing.



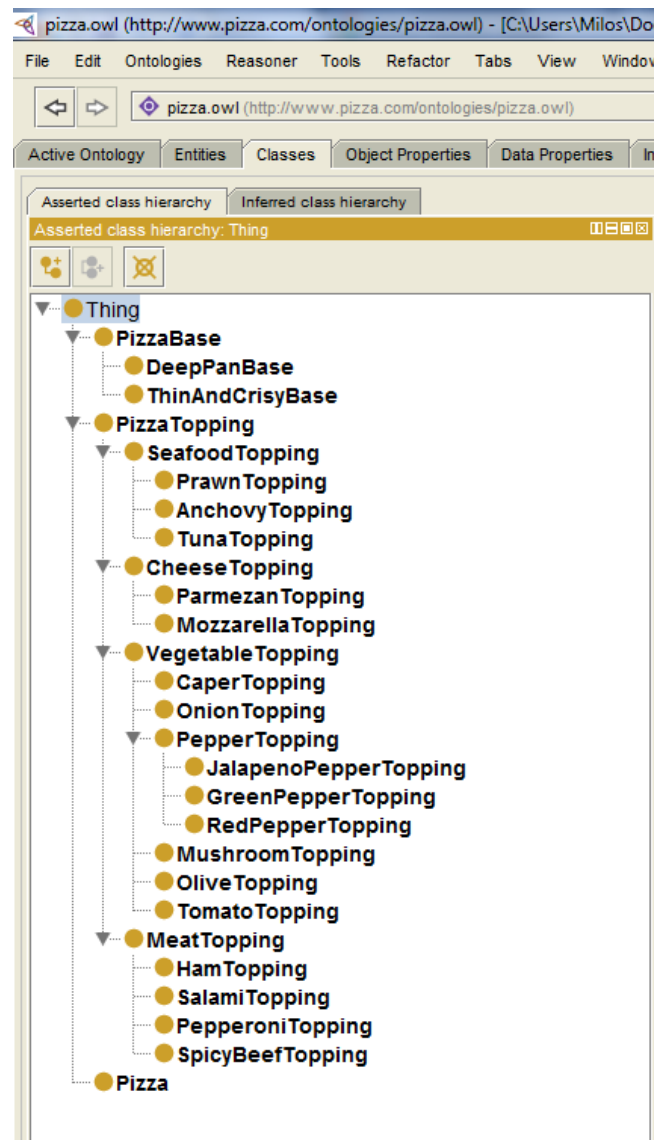
Слика 1

3. Дефинирајте ги трите класи Pizza, PizzaTopping и PizzaBase како дисјунктни (disjoint). За оваа цел селектирајте една од класите и до десниот дел од интерфејсот изберете “Disjoint classes”, со клик на знакот „+“. За секоја од класите дефинирајте ги останатите две како дисјунктни со неа.

Напомена: за побрзо дефинирање на дисјунктни класи (особено кога нивниот број е голем) направете multiple селекција на сите останати класи. На тој начин сите ќе станат дисјунктни.

4. Креирајте ги класите ThinAndCrispyBase и DeepPanBase како подкласи на PizzaBase. Дефинирајте ги овие две класи како дисјунктни.
5. Креирајте ги подкласите MeatTopping, VegetableTopping, CheeseTopping и SeafoodTopping како подкласи на PizzaTopping. Дефинирајте ги овие класи како дисјунктни.

6. Во рамките на класата MeatTopping додадете ги следниве дисјунктни подкласи: SpicyBeefTopping, PepperoniTopping, SalamiTopping и HamTopping.
7. Во рамките на класата VegetableTopping додадете ги следниве дисјунктни подкласи: TomatoTopping, OliveTopping, MushroomTopping, PepperTopping, OnionTopping и CaperTopping.
8. Во рамките на класата PepperTopping додадете ги следниве дисјунктни подкласи: RedPepperTopping, GreenPepperTopping и JalapenoPepperTopping.
9. Во рамките на класата CheeseTopping додадете ги следниве дисјунктни подкласи: MozzarellaTopping и ParmezanTopping.
10. Во рамките на класата SeafoodTopping додадете ги следниве дисјунктни подкласи: TunaTopping, AnchovyTopping и PrawnTopping.
11. После овие точки, вашата онтологија треба да изгледа како на Слика 2. Доколку имате грешки, вратете се назад на соодветниот чекор и отстранете ги.



Слика 2

Слики од решението на I. Дефинирање на класите во онтологијата:

The screenshot shows an ontology editor interface. The top navigation bar includes tabs for 'Active ontology', 'Entities', 'Classes', 'Object properties', 'Data properties', 'Individuals by class', and 'DL Query'. The 'Classes' tab is active, and the 'Class hierarchy' sub-tab is selected. The left pane displays a class hierarchy starting with 'owl:Thing' as the root. Under 'owl:Thing', there are three classes: 'Pizza', 'PizzaBase', and 'PizzaTopping'. The 'Pizza' class is highlighted. The right pane shows the 'Annotations' tab for the 'Pizza' class. It lists various annotation types with plus icons for expansion: 'Equivalent To', 'SubClass Of', 'General class axioms', 'SubClass Of (Anonymous Ancestor)', 'Instances', 'Target for Key', and 'Disjoint With'. Under 'Disjoint With', two classes are listed: 'PizzaTopping' and 'PizzaBase'.

The screenshot shows the same ontology editor interface, but with the 'SeafoodTopping' class selected. The left pane shows a more detailed class hierarchy. Under 'owl:Thing', there is 'Pizza', which has a sub-class 'PizzaBase'. Under 'PizzaBase', there are 'DeepPanBase' and 'ThinAndCrispyBase'. Below these is 'PizzaTopping', which has three sub-classes: 'SeafoodTopping' (highlighted), 'CheeseTopping', and 'VegetableTopping'. The right pane shows the 'Annotations' tab for 'SeafoodTopping'. It lists the same annotation types as the previous screenshot. Under 'SubClass Of', 'PizzaTopping' is listed. Under 'Disjoint With', 'VegetableTopping', 'MeatTopping', and 'CheeseTopping' are listed.

Active ontology x Entities x **Classes** x Object properties x Data properties x Individuals by class x DL Query x

Class hierarchy Class hierarchy (inferred)

Class hierarchy: CaperTopping Annotations Usage

Annotations: CaperTopping

Annotations +

Description: CaperTopping

SubClass Of +

- VegetableTopping

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

- TomatoTopping
- OnionTopping
- PepperTopping
- MushroomTopping
- OliveTopping

Active ontology x Entities x **Classes** x Object properties x Data properties x Individuals by class x DL Query x

Class hierarchy Class hierarchy (inferred)

Class hierarchy: owl:Thing Annotations Usage

Annotations: owl:Thing

Annotations +

Description: owl:Thing

Equivalent To +

SubClass Of +

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

Disjoint Union Of +

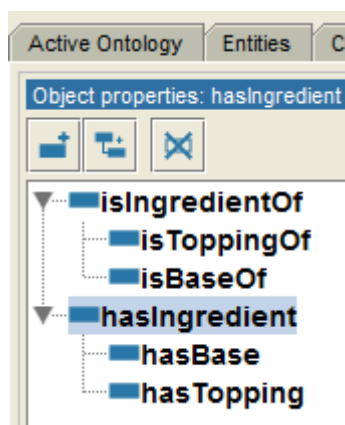
II. Дефинирање на релациите (својствата) во онтологијата

12. Префрлете се во табот Object Properties. Кликнете на копчето Add Property и додадете објектно својство со име hasIngredient.
13. Селектирајте го својството hasIngredient. Додадете му ги подсвојствата hasTopping и hasBase.
14. Креирајте ново објектно својство, isIngredientOf. Во десниот дел од интерфејсот додадете информација дека ова својство е инверзно на својството hasIngredient. За таа цел искористете го копчето „+“ до опцијата Inverse properties. Со ова дефинираме дека својствата hasIngredient и isIngredientOf се инверзни меѓу себе.
15. Креирајте ги својствата isBaseOf и isToppingOf како подсвојства на isIngredientOf. Дефинирајте ги како инверзни својства на hasBase и hasTopping, соодветно.
16. Селектирајте го објектното својство hasIngredient. Дефинирајте го како транзитивно својство, преку селектирање на check box-от даден во Characteristics (во средина на интерфејсот).
17. Дефинирајте го и својството isIngredientOf како транзитивно.
18. Дефинирајте го својството hasBase како функционално својство. Што значи одредено својство да биде функционално?
19. Специфицирајте ги доменот и опсегот на својството hasTopping. Доменот ги одредуваше класите кои може да ги имаат овие релации, а опсегот ги одредува класите кои можат да бидат вредности на овие релации.

Кликнете на „+“ кај Domains (intersection) во делот Description (во десниот дел од интерфејсот). Во прозорецот кој ќе ви се појави, изберете го табот “Asserted class hierarchy”. Оттаму изберете ја класата Pizza како домен на својството hasTopping.

Аналогно на овие постапки, дефинирајте ја класата PizzaTopping како опсег (range) на својството hasTopping.

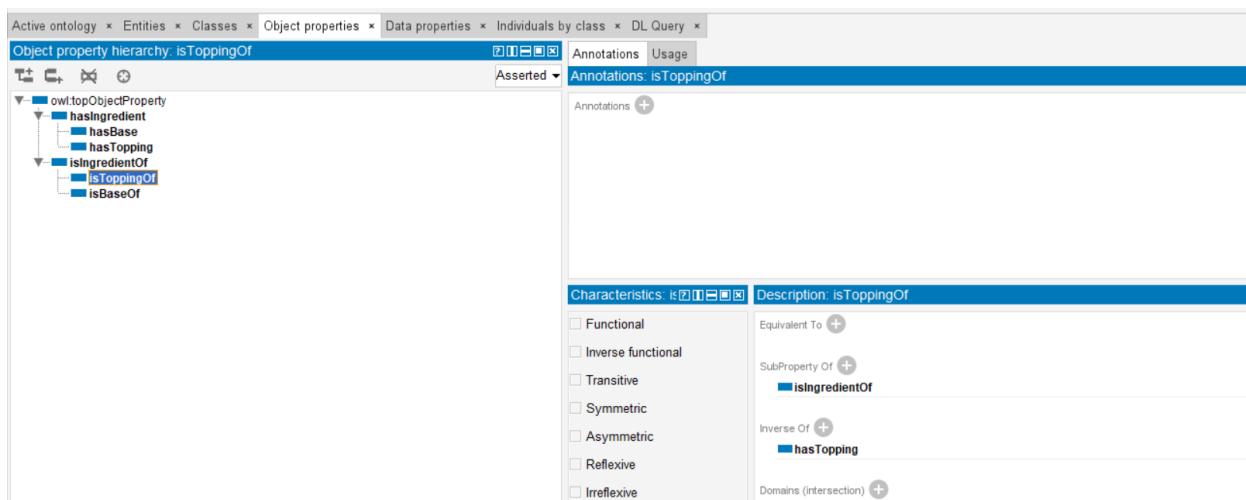
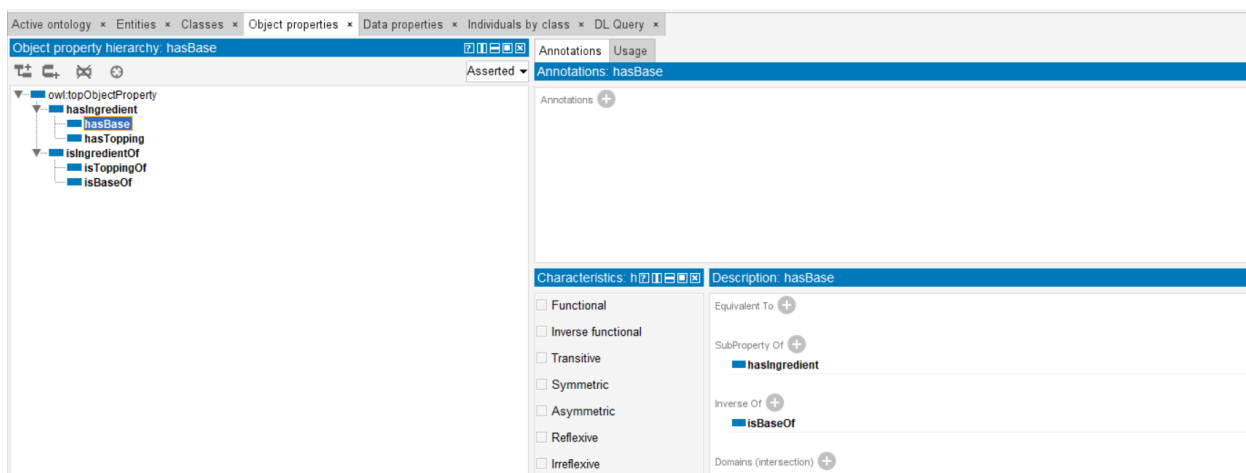
Ова означува дека својството hasTopping се однесува на инстанци од класата Pizza (доменот), а како вредност може да има инстанци од класата PizzaTopping (опсегот).
20. Специфицирајте ги доменот и опсегот на својството isToppingOf. Поради тоа што својството е инверзно со hasTopping, доменот и опсегот на isToppingOf имаат обратни вредности од вредностите кај hasTopping својството. Тоа значи дека доменот на isToppingOf е PizzaTopping, додека пак опсегот е Pizza.
21. Специфицирајте ги доменот и опсегот на својството hasBase и на неговото инверзно својство isBaseOf. Домен на hasBase е Pizza, а опсег е PizzaBase. Кај isBaseOf доменот и опсегот се обратни, односно PizzaBase е доменот, а Pizza е опсегот.



Слика 3

22. Доколку правилно сте ги креирале релациите (својствата) во онтологијата, би требало да добиете изглед како на Слика 3.

Слики од решението на II. Дефинирање на релациите (својствата) во онтологијата:



Active ontology * Entities * Classes * Object properties * Data properties * Individuals by class * DL Query *

Object property hierarchy: hasIngredient Annotations Usage

Annotations: hasIngredient

Annotations +

Characteristics: h i f t s a

☐ Functional
☐ Inverse functional
☒ Transitive
☐ Symmetric
☐ Asymmetric
☐ Reflexive
☐ Irreflexive

Description: hasIngredient

Equivalent To +

SubProperty Of +

Inverse Of +

isIngredientOf

Domains (intersection) +

Active ontology * Entities * Classes * Object properties * Data properties * Individuals by class * DL Query *

Object property hierarchy: hasBase Annotations Usage

Annotations: hasBase

Annotations +

Characteristics: h i f t s a

☒ Functional
☐ Inverse functional
☐ Transitive
☐ Symmetric
☐ Asymmetric
☐ Reflexive
☐ Irreflexive

Description: hasBase

Equivalent To +

SubProperty Of +

hasIngredient

Inverse Of +

isBaseOf

Domains (intersection) +

Active ontology * Entities * Classes * Object properties * Data properties * Individuals by class * DL Query *

Object property hierarchy: hasTopping Annotations Usage

Annotations: hasTopping

Annotations +

Characteristics: h i f t s a

☐ Functional
☐ Inverse functional
☐ Transitive
☐ Symmetric
☐ Asymmetric
☐ Reflexive
☐ Irreflexive

Description: hasTopping

Equivalent To +

SubProperty Of +

hasIngredient

Inverse Of +

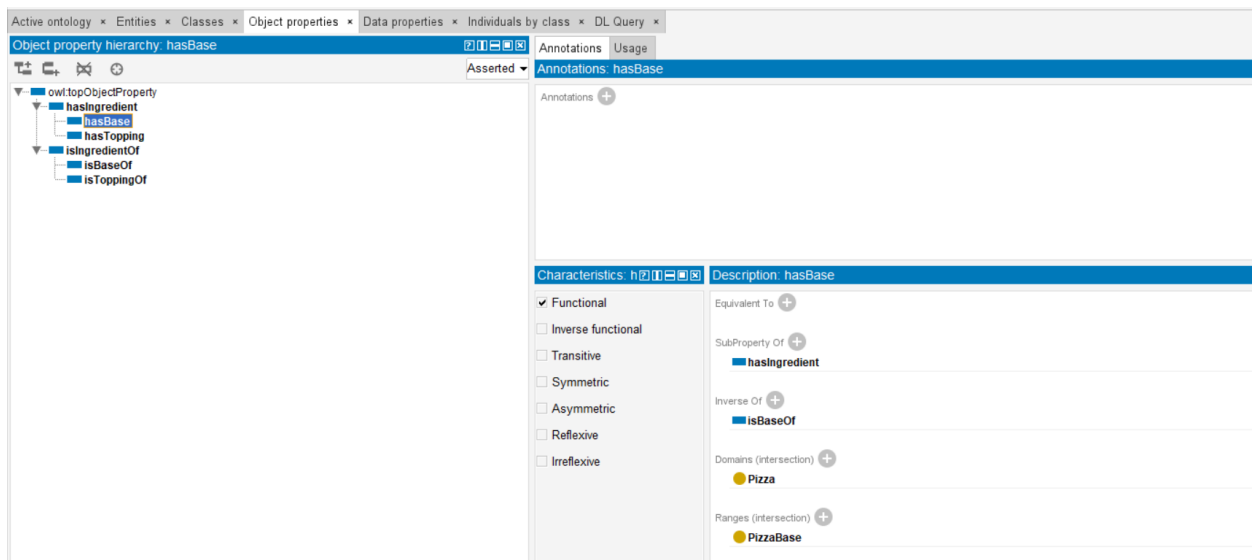
isToppingOf

Domains (intersection) +

Pizza

Ranges (intersection) +

PizzaTopping



III. Дефинирање ограничувања во онтологијата

23. Дефинирањето на ограничувања за класите во една онтологија се прави со дефинирање на анонимни суперкласи. Инстанците од класата ќе мора да ги почитуваат овие ограничувања, за да бидат деца и на суперкласата.

Вратете се назад на табот Classes и за класата Pizza дефинирајте ограничување дека нејзината релација hasBase треба да има барем една вредност (some) од PizzaBase. За да го направите ова, мора да додадете нова анонимна суперкласа за класата Pizza. Кликнете на „+“ веднаш до Superclasses. Од прозорецот одберете го табот “Object restriction creator”, во кој ќе дефинирате дека својството hasBase има тип на рестрикција Some (existential) кон класата PizzaBase. Кликнете ОК.

На овој начин, со помош на анонимна суперкласа дефинирајте ограничувања за класата Pizza.

24. Во класата Pizza дефинирајте подкласа NamedPizza. Во NamedPizza дефинирајте подкласа MargheritaPizza. Додајте го следниот коментар кај MargheritaPizza: „A pizza that only has Mozzarella and Tomato toppings“. Додавањето на коментар се прави со својството comment, во делот Annotations +.
25. Каква вредност за „Inferred anonymous superclasses“ има класата MargheritaPizza? Зошто?
26. На сличен начин како кај точка 22, дефинирајте ограничувања за MargheritaPizza дека својството hasTopping мора да има someValueFrom од MozzarellaTopping. Каков запис има во полето Superclasses сега за MargheritaPizza?

Дефинирајте ограничување дека својството hasTopping мора да има и someValueFrom од TomatoTopping.

27. Креирајте нова класа AmericanPizza, преку клонирање на класата MargheritaPizza. Тоа може да се направи со селектирање на MargheritaPizza и одбирање на опцијата “Edit – Duplicate selected class”. Кај AmericanPizza дефинирајте уште едно дополнително ограничување: додадете PepperoniTopping.
28. Дефинирајте ги AmericanPizza и MargheritaPizza како дисјунктни класи.

Слики од решението на III. Дефинирање ограничувања во онтологијата:

The screenshot shows the Protégé interface with the 'Classes' tab selected. The class hierarchy on the left shows 'Pizza' as a subclass of 'NamedPizza' and 'PizzaBase'. The 'Annotations' panel on the right shows the following information for 'Pizza':

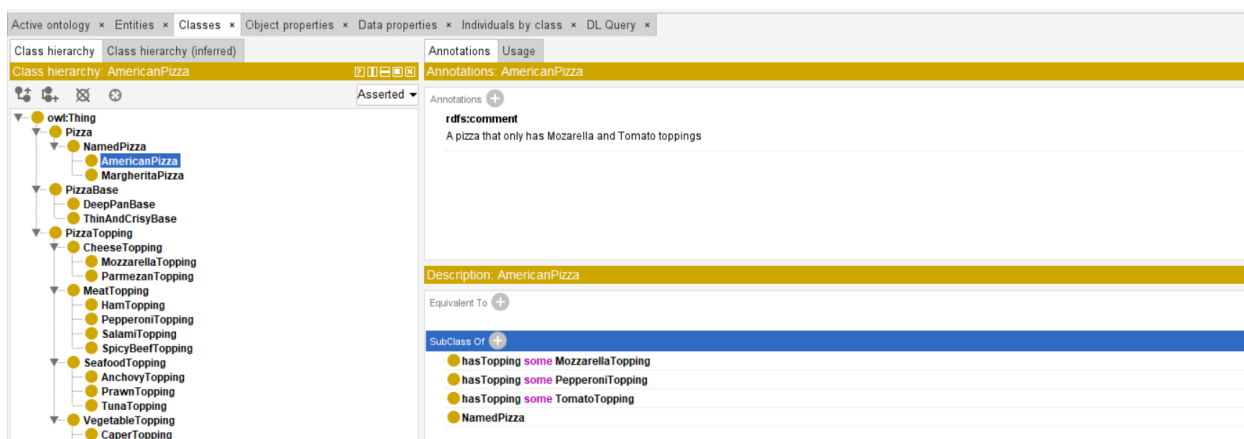
- Description:** Pizza
- Equivalent To:** (empty)
- SubClass Of:** (empty)
- General class axioms:** (empty)
- SubClass Of (Anonymous Ancestor):**
 - hasBase some PizzaBase
- Instances:** (empty)
- Target for Key:** (empty)
- Disjoint With:**
 - PizzaTopping
 - PizzaBase

The screenshot shows the Protégé interface with the 'Classes' tab selected. The class hierarchy on the left shows 'MargheritaPizza' as a subclass of 'NamedPizza' and 'PizzaBase'. The 'Annotations' panel on the right shows the following information for 'MargheritaPizza':

- Description:** MargheritaPizza
- Equivalent To:** (empty)
- SubClass Of:**
 - NamedPizza
- General class axioms:** (empty)
- SubClass Of (Anonymous Ancestor):**
 - hasBase some PizzaBase

The screenshot shows the Protégé interface with the 'Classes' tab selected. The class hierarchy on the left shows 'MargheritaPizza' as a subclass of 'NamedPizza' and 'PizzaBase'. The 'Annotations' panel on the right shows the following information for 'MargheritaPizza':

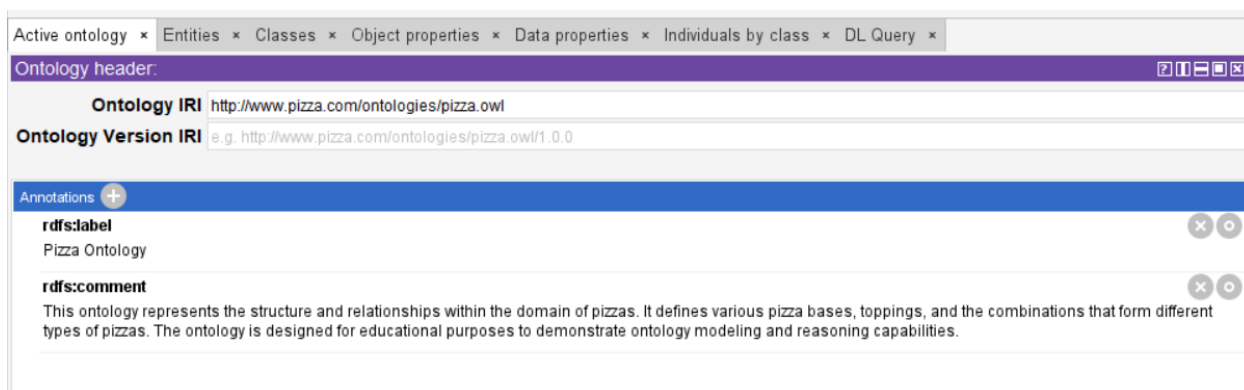
- Description:** MargheritaPizza
- Equivalent To:** (empty)
- SubClass Of:**
 - hasTopping some MozzarellaTopping
 - hasTopping some TomatoTopping
 - NamedPizza



IV. Дефинирање на детали за самата онтологија

29. Отворете го табот Active Ontology. Преку Annotations опцијата, додадете коментар (comment) со кој ќе ја опишете онтологијата и лабела (label) која ќе го означи името на самата онтологија (Pizza Ontology).

Слики од решението на IV. Дефинирање на детали за самата онтологија:



V. Визуелизација на онтологијата

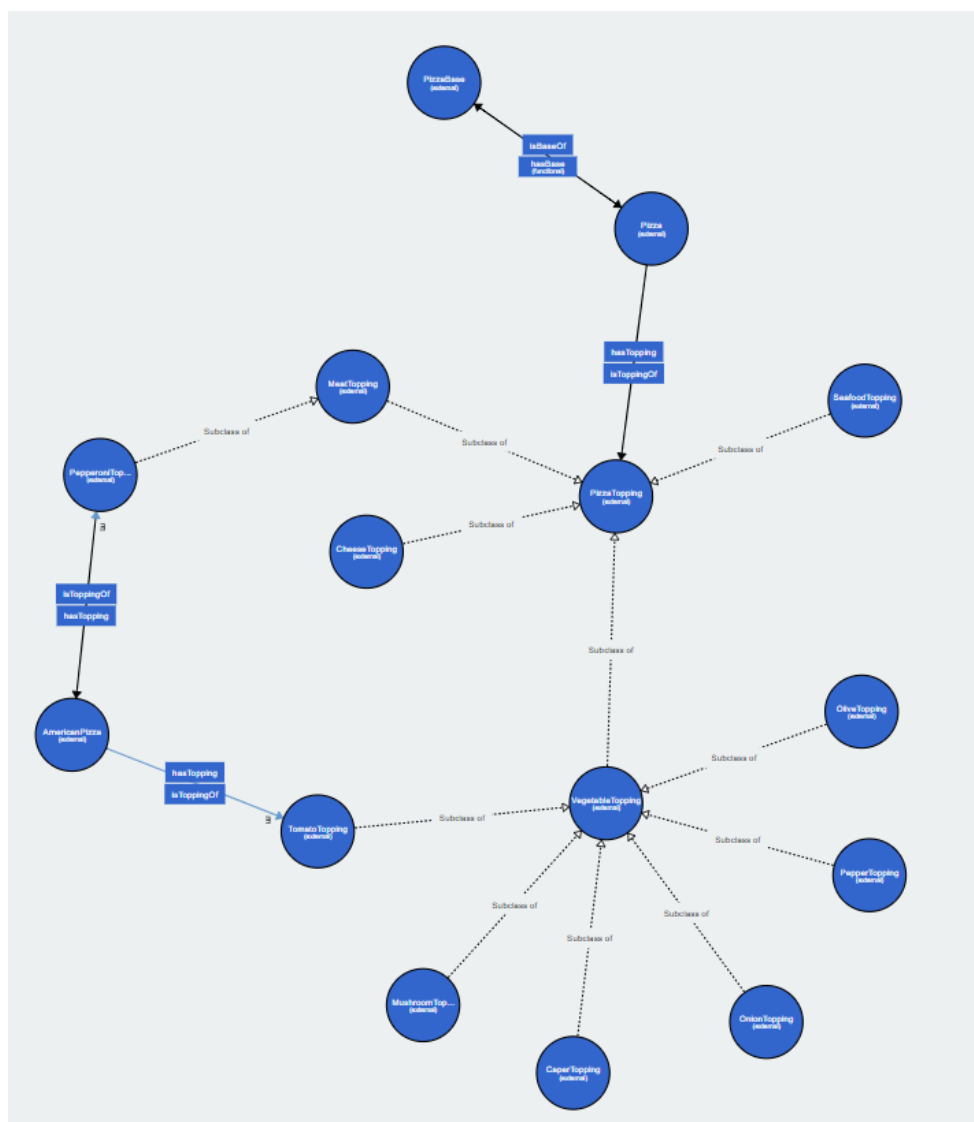
[ProtégéVOWL](#) е додаток за Protégé кој овозможува визуелизација на онтологии, во форма на граф. За визуелизација, тој ја користи [VOWL v2](#) визуелната нотација за исцртување OWL онтологии.

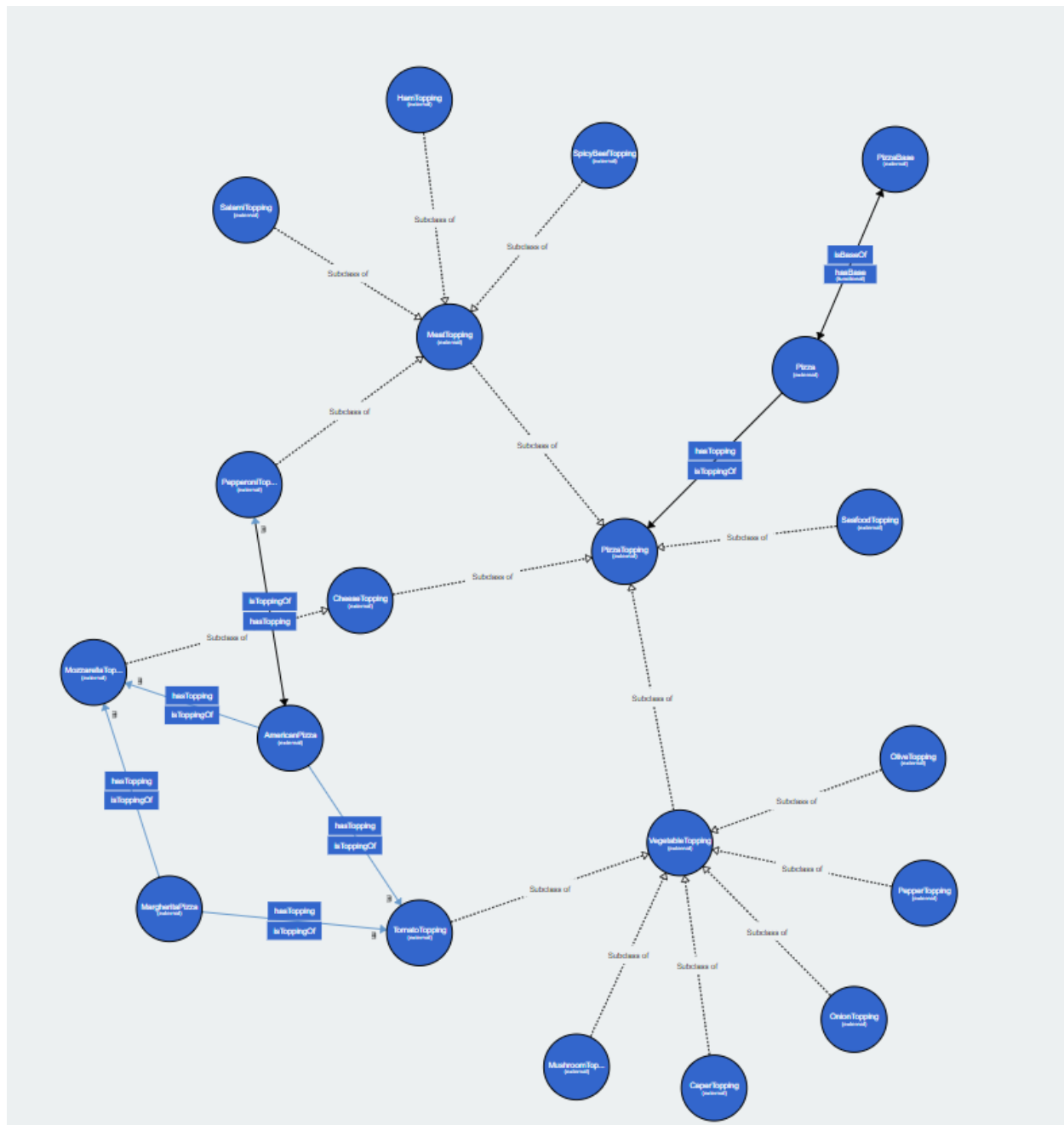
Преземете го [ProtégéVOWL](#) додатокот и инсталирајте го според упатството на неговата веб страна.

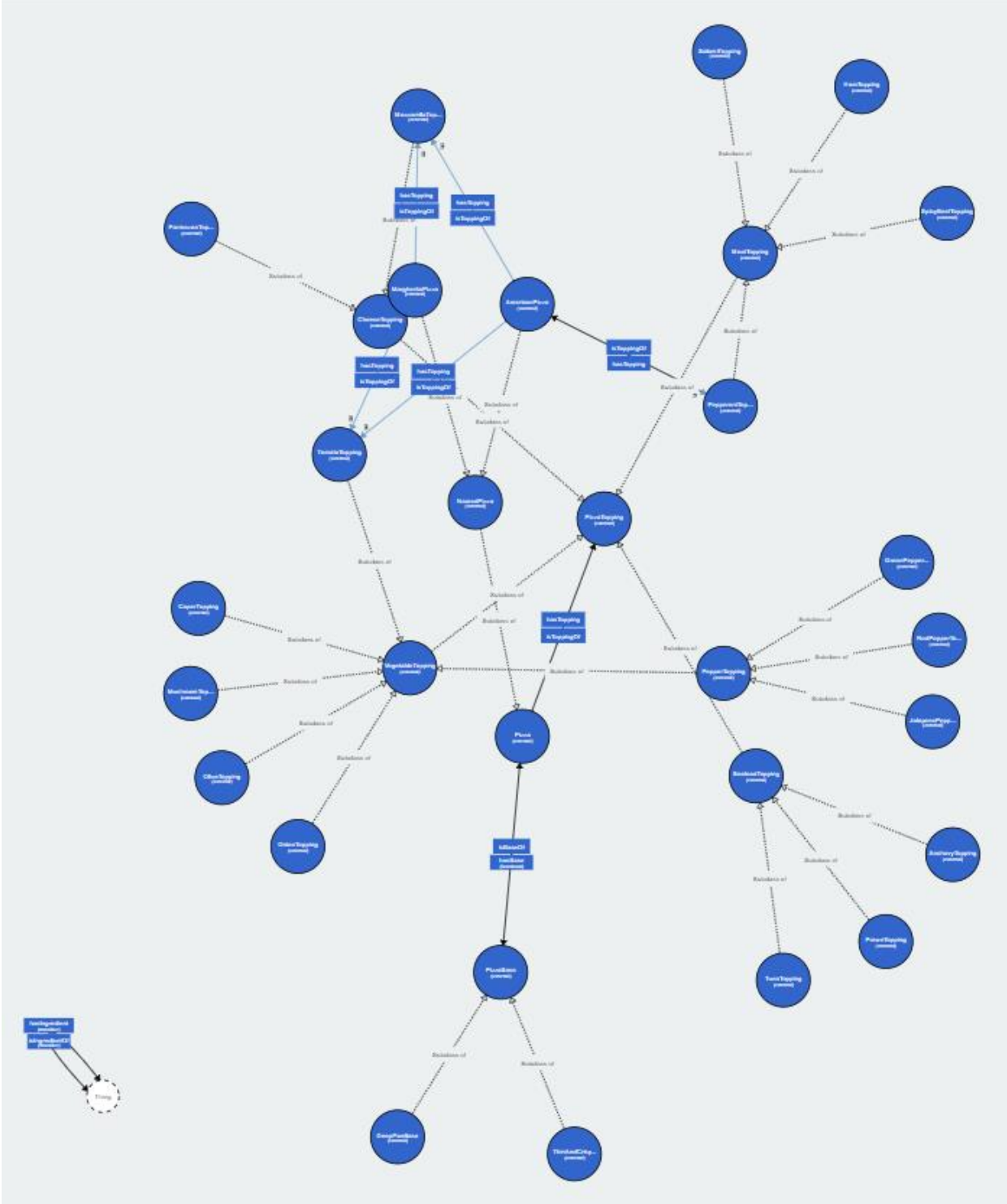
30. Отворете го табот VOWL. Пробајте да ги разместите класите од онтологијата така што нивната хиерархиска структура ќе биде јасно видлива.

31. Разгледајте ги класите и нивната поврзаност со релации. Дали структурата одговара на она што го очекувавте додека ја креиравте онтологијата?

Слики од решението на *V. Визуелизација на онтологијата*:







VI. Онтологијата како Turtle датотека

32. По default, онтологијата е креирана со OWL/XML синтакса. Искористете ја опцијата за избор на формат на онтологијата при Save As... и снимете ја онтологијата и како Turtle датотека (Pizza.ttl).
33. Отворете ја Turtle датотеката во текстуален едитор и разгледајте ги креираните тројки со кои е дефинирана онтологијата од вежбата.

Слики од решението на VI. Онтологијата како Turtle датотека:

```
@prefix : <http://www.semanticweb.org/pc/ontologies/2024/10/untitled-ontology-16/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://www.semanticweb.org/pc/ontologies/2024/10/untitled-ontology-16/> .

<http://www.pizza.com/ontologies/pizza.owl> rdf:type owl:Ontology ;
      rdfs:comment "This ontology represents the structure and
relationships within the domain of pizzas. It defines various pizza bases, toppings, and the
combinations that form different types of pizzas. The ontology is designed for educational
purposes to demonstrate ontology modeling and reasoning capabilities." ;
      rdfs:label "Pizza Ontology" .

#####
#   Object Properties
#####

### http://www.pizza.com/ontologies/pizza.owl#hasBase
:hasBase rdf:type owl:ObjectProperty ;
      rdfs:subPropertyOf :hasIngredient ;
      owl:inverseOf :isBaseOf ;
      rdf:type owl:FunctionalProperty ;
      rdfs:domain :Pizza ;
      rdfs:range :PizzaBase .

### http://www.pizza.com/ontologies/pizza.owl#hasIngredient
:hasIngredient rdf:type owl:ObjectProperty ;
      owl:inverseOf :isIngredientOf ;
      rdf:type owl:TransitiveProperty .

### http://www.pizza.com/ontologies/pizza.owl#hasTopping
```

```
:hasTopping rdf:type owl:ObjectProperty ;  
  rdfs:subPropertyOf :hasIngredient ;  
  owl:inverseOf :isToppingOf ;  
  rdfs:domain :Pizza ;  
  rdfs:range :PizzaTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#isBaseOf  
:isBaseOf rdf:type owl:ObjectProperty ;  
  rdfs:subPropertyOf :isIngredientOf ;  
  rdfs:domain :PizzaBase ;  
  rdfs:range :Pizza .
```

```
### http://www.pizza.com/ontologies/pizza.owl#isIngredientOf  
:isIngredientOf rdf:type owl:ObjectProperty ,  
  owl:TransitiveProperty .
```

```
### http://www.pizza.com/ontologies/pizza.owl#isToppingOf  
:isToppingOf rdf:type owl:ObjectProperty ;  
  rdfs:subPropertyOf :isIngredientOf ;  
  rdfs:domain :PizzaTopping ;  
  rdfs:range :Pizza .
```

```
#####  
# Classes  
#####
```

```
### http://www.pizza.com/ontologies/pizza.owl#AmericanPizza  
:AmericanPizza rdf:type owl:Class ;  
  rdfs:subClassOf :NamedPizza ,  
    [ rdf:type owl:Restriction ;  
      owl:onProperty :hasTopping ;  
      owl:someValuesFrom :MozzarellaTopping  
    ] ,  
    [ rdf:type owl:Restriction ;  
      owl:onProperty :hasTopping ;  
      owl:someValuesFrom :PepperoniTopping  
    ] ,  
    [ rdf:type owl:Restriction ;  
      owl:onProperty :hasTopping ;  
      owl:someValuesFrom :TomatoTopping  
    ] ;  
  owl:disjointWith :MargheritaPizza ;  
  rdfs:comment "A pizza that only has Mozzarella and Tomato toppings" .
```

```
### http://www.pizza.com/ontologies/pizza.owl#AnchovyTopping  
:AnchovyTopping rdf:type owl:Class ;  
  rdfs:subClassOf :SeafoodTopping ;  
  owl:disjointWith :PrawnTopping ,
```

:TunaTopping .

```
### http://www.pizza.com/ontologies/pizza.owl#CaperTopping
:CaperTopping rdf:type owl:Class ;
  rdfs:subClassOf :VegetableTopping ;
  owl:disjointWith :MushroomTopping ,
    :OliveTopping ,
    :OnionTopping ,
    :PepperTopping ,
    :TomatoTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#CheeseTopping
:CheeseTopping rdf:type owl:Class ;
  rdfs:subClassOf :PizzaTopping ;
  owl:disjointWith :MeatTopping ,
    :SeafoodTopping ,
    :VegetableTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#DeepPanBase
:DeepPanBase rdf:type owl:Class ;
  rdfs:subClassOf :PizzaBase ;
  owl:disjointWith :ThinAndCrispyBase .
```

```
### http://www.pizza.com/ontologies/pizza.owl#GreenPepperTopping
:GreenPepperTopping rdf:type owl:Class ;
  rdfs:subClassOf :PepperTopping ;
  owl:disjointWith :JalapenoPepperTopping ,
    :RedPepperTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#HamTopping
:HamTopping rdf:type owl:Class ;
  rdfs:subClassOf :MeatTopping ;
  owl:disjointWith :PepperoniTopping ,
    :SalamiTopping ,
    :SpicyBeefTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#JalapenoPepperTopping
:JalapenoPepperTopping rdf:type owl:Class ;
  rdfs:subClassOf :PepperTopping ;
  owl:disjointWith :RedPepperTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#MargheritaPizza
:MargheritaPizza rdf:type owl:Class ;
  rdfs:subClassOf :NamedPizza ,
    [ rdf:type owl:Restriction ;
      owl:onProperty :hasTopping ;
      owl:someValuesFrom :MozzarellaTopping
    ] ,
```

```
[ rdf:type owl:Restriction ;  
  owl:onProperty :hasTopping ;  
  owl:someValuesFrom :TomatoTopping  
] ;  
rdfs:comment "A pizza that only has Mozzarella and Tomato toppings" .
```

```
### http://www.pizza.com/ontologies/pizza.owl#MeatTopping  
:MeatTopping rdf:type owl:Class ;  
  rdfs:subClassOf :PizzaTopping ;  
  owl:disjointWith :SeafoodTopping ,  
    :VegetableTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#MozzarellaTopping  
:MozzarellaTopping rdf:type owl:Class ;  
  rdfs:subClassOf :CheeseTopping ;  
  owl:disjointWith :ParmezanTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#MushroomTopping  
:MushroomTopping rdf:type owl:Class ;  
  rdfs:subClassOf :VegetableTopping ;  
  owl:disjointWith :OliveTopping ,  
    :OnionTopping ,  
    :PepperTopping ,  
    :TomatoTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#NamedPizza  
:NamedPizza rdf:type owl:Class ;  
  rdfs:subClassOf :Pizza .
```

```
### http://www.pizza.com/ontologies/pizza.owl#OliveTopping  
:OliveTopping rdf:type owl:Class ;  
  rdfs:subClassOf :VegetableTopping ;  
  owl:disjointWith :OnionTopping ,  
    :PepperTopping ,  
    :TomatoTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#OnionTopping  
:OnionTopping rdf:type owl:Class ;  
  rdfs:subClassOf :VegetableTopping ;  
  owl:disjointWith :PepperTopping ,  
    :TomatoTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#ParmezanTopping  
:ParmezanTopping rdf:type owl:Class ;  
  rdfs:subClassOf :CheeseTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#PepperTopping
:PepperTopping rdf:type owl:Class ;
    rdfs:subClassOf :VegetableTopping ;
    owl:disjointWith :TomatoTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#PepperoniTopping
:PepperoniTopping rdf:type owl:Class ;
    rdfs:subClassOf :MeatTopping ;
    owl:disjointWith :SalamiTopping ,
        :SpicyBeefTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#Pizza
:Pizza rdf:type owl:Class ;
    owl:disjointWith :PizzaBase ,
        :PizzaTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#PizzaBase
:PizzaBase rdf:type owl:Class ;
    owl:disjointWith :PizzaTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#PizzaTopping
:PizzaTopping rdf:type owl:Class .
```

```
### http://www.pizza.com/ontologies/pizza.owl#PrawnTopping
:PrawnTopping rdf:type owl:Class ;
    rdfs:subClassOf :SeafoodTopping ;
    owl:disjointWith :TunaTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#RedPepperTopping
:RedPepperTopping rdf:type owl:Class ;
    rdfs:subClassOf :PepperTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#SalamiTopping
:SalamiTopping rdf:type owl:Class ;
    rdfs:subClassOf :MeatTopping ;
    owl:disjointWith :SpicyBeefTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#SeafoodTopping
:SeafoodTopping rdf:type owl:Class ;
    rdfs:subClassOf :PizzaTopping ;
    owl:disjointWith :VegetableTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#SpicyBeefTopping
```

```
:SpicyBeefTopping rdf:type owl:Class ;  
    rdfs:subClassOf :MeatTopping .
```

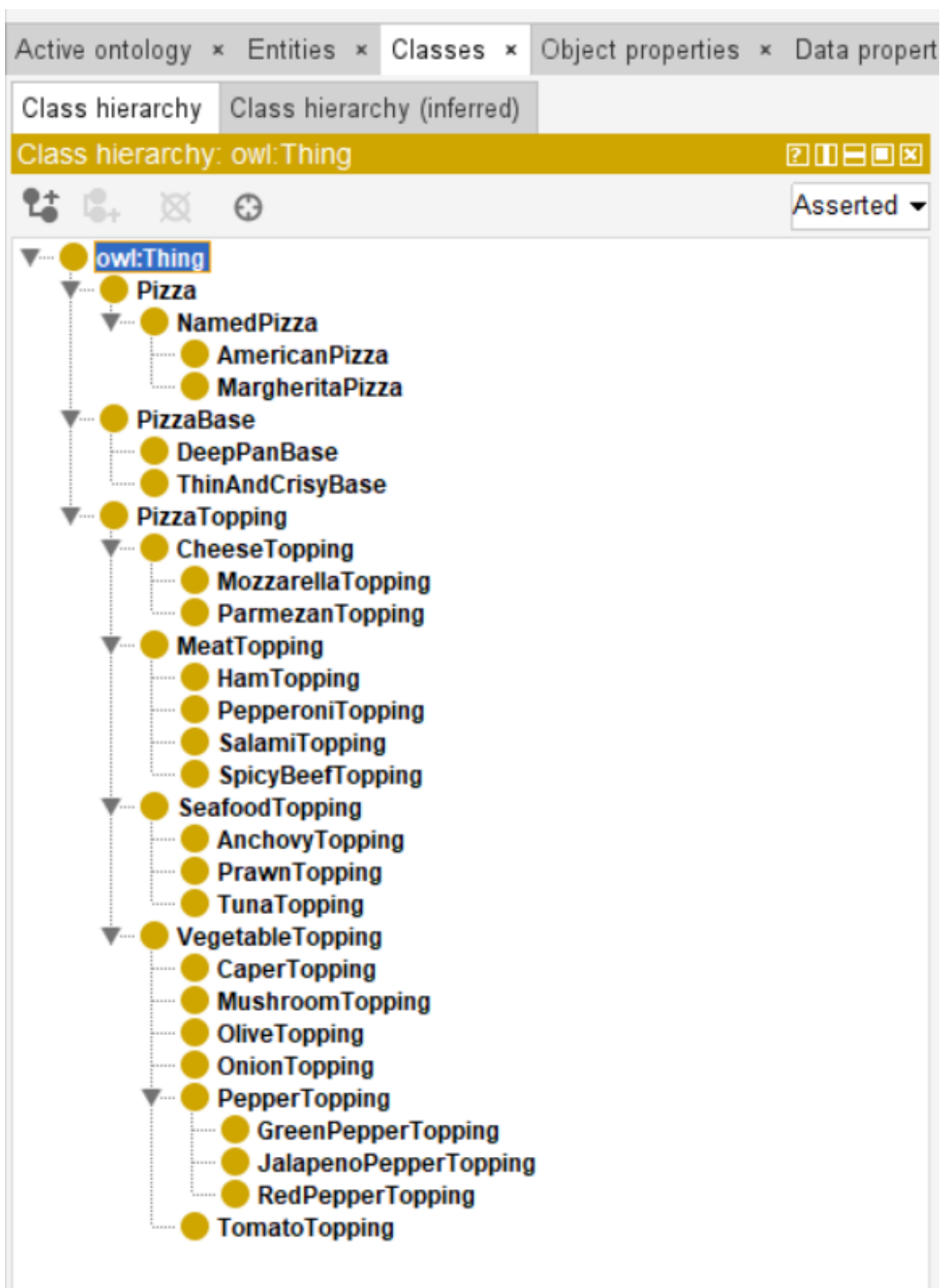
```
### http://www.pizza.com/ontologies/pizza.owl#ThinAndCrisyBase  
:ThinAndCrisyBase rdf:type owl:Class ;  
    rdfs:subClassOf :PizzaBase .
```

```
### http://www.pizza.com/ontologies/pizza.owl#TomatoTopping  
:TomatoTopping rdf:type owl:Class ;  
    rdfs:subClassOf :VegetableTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#TunaTopping  
:TunaTopping rdf:type owl:Class ;  
    rdfs:subClassOf :SeafoodTopping .
```

```
### http://www.pizza.com/ontologies/pizza.owl#VegetableTopping  
:VegetableTopping rdf:type owl:Class ;  
    rdfs:subClassOf :PizzaTopping .
```

```
### http://www.w3.org/2002/07/owl#Thing  
owl:Thing rdfs:subClassOf [ rdf:type owl:Restriction ;  
    owl:onProperty :hasBase ;  
    owl:someValuesFrom :PizzaBase  
] .
```



Изработил:
Бојан Ристов (211151)