

**Домашно бр. 3
OWL**

203235 - Димитрија Тимески

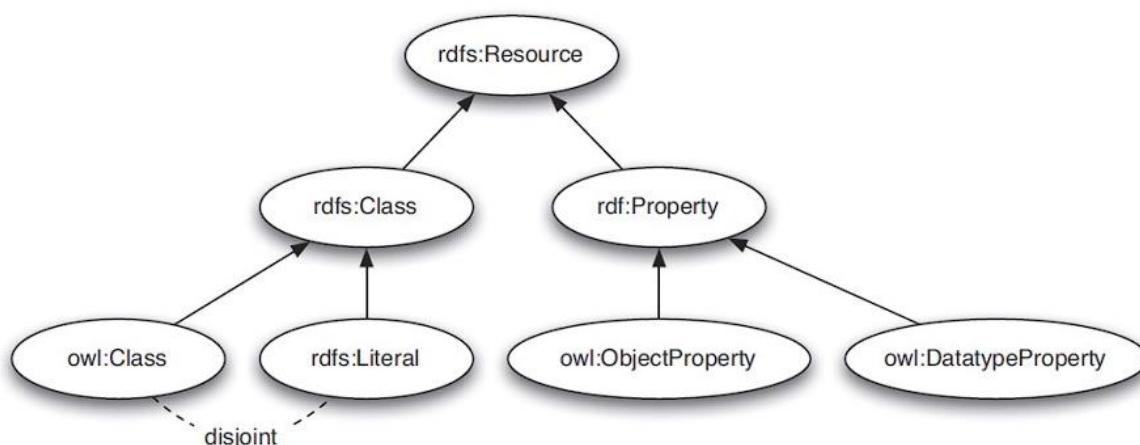
15-Nov-2023

A) Домашна задача

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

1. Онтологија е формален начин на дефинирање на домени од реалноста
2. Јазик за опис на онтологија има за цел да дефинира шема и структура преку кои се дефинираат:
 - Класи
 - Својства
 - Релации

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



OWL е изграден врз RDF/RDFS и тој ги поседува сите негови концепти.

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

Постојат 2 типа на својства во OWL:

1. owl:ObjectProperty
 - Се однесува на индивидуи, т.е. класи
2. owl:DatatypeProperty
 - Се однесува на литерали (string, numbers, dates)

- 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/ontology#> .
```

```
:Person rdf:type owl:Class .  
:Employee rdf:type owl:Class ;  
    rdfs:subClassOf :Person .
```

```
:hasEmployee rdf:type owl:ObjectProperty ;  
    owl:inverseOf :isEmployedBy .
```

```
:isEmployedBy rdf:type owl:ObjectProperty .
```

```
:John rdf:type :Person .  
:Company rdf:type :Organization .
```

```
:John :isEmployedBy :Company .
```

➤ Инверзно на “hasEmployee” е “isEmployedBy”

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

Тие се дефинираат преку класи со тоа што и се ограничуваат можните инстанци преку нивните вредности.

OWL прави разлика помеѓу следниве две:

- Ограничивање на вредноста
- Ограничивање на кардиналноста

```
Class: Person  
    EquivalentTo: hasName exactly 1 xsd:string
```

```
Class: Person  
    SubClassOf: hasAge some xsd:integer
```

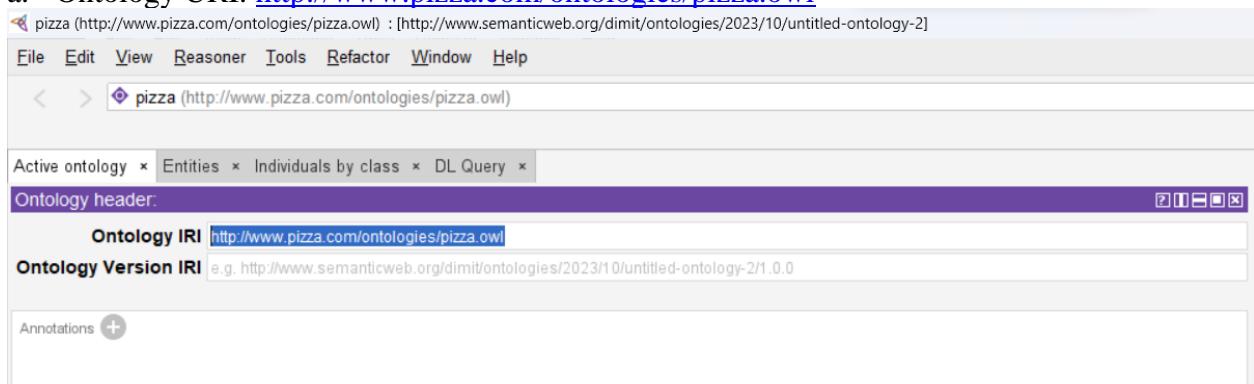
Б) Практична задача

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, класификацији, шеми на бази на податоци, итн.

- Стартувајте го Protégé. Креирајте нова OWL онтологија:

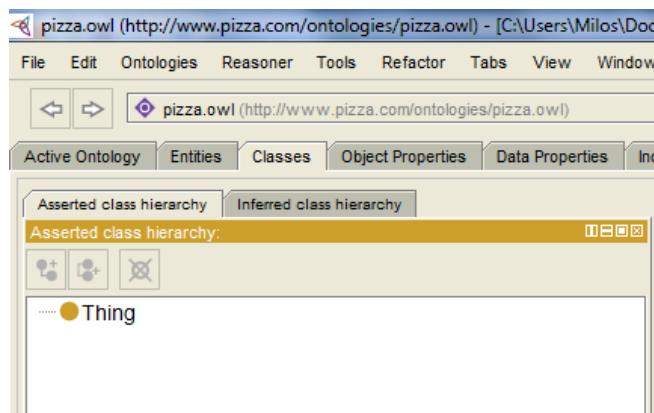
- Ontology URI: <http://www.pizza.com/ontologies/pizza.owl>



- Локација: MyDocuments/WBS/pizza.owl

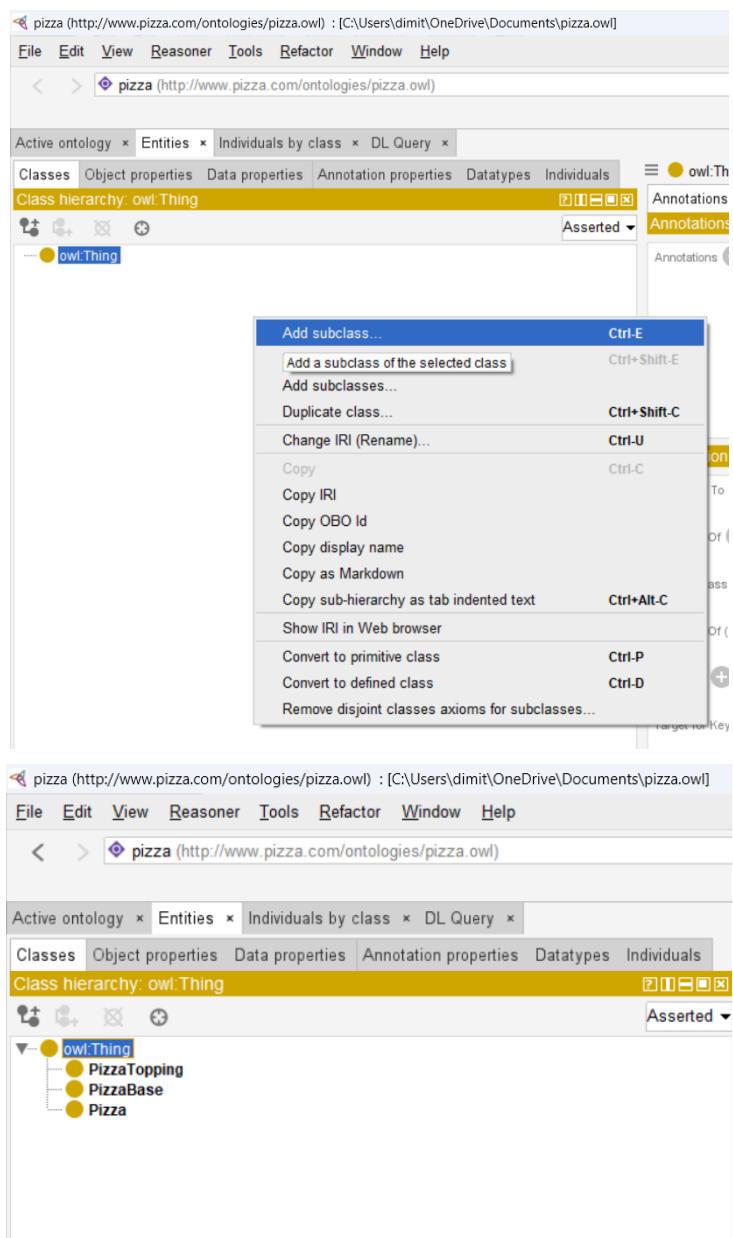
I. Дефинирање на класите во онтологијата

- 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

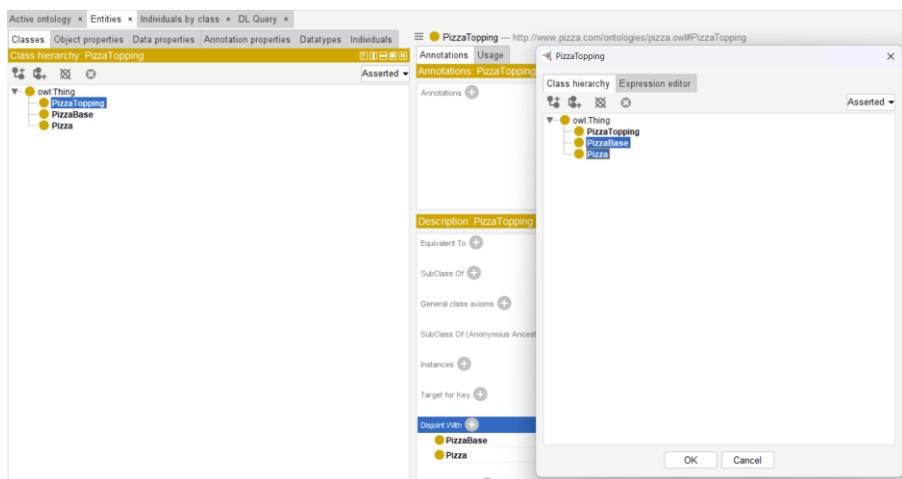
Веб базирани системи 2023



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.

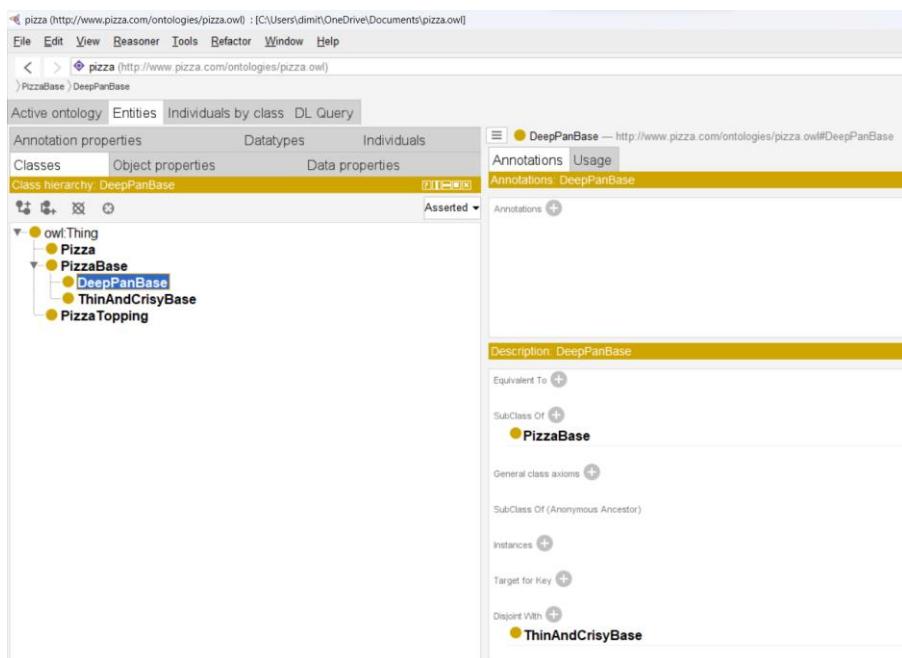
Напомена: за побрзо дефинирање на дисјунктни класи (особено кога нивниот број е голем) направете multiple селекција на сите останати класи. На тој начин сите ќе станат дисјунктни.

Веб базирани системи 2023



Ова чекор го повторив и за другите две.

- Креирајте ги класите ThinAndCrisyBase и DeepPanBase како подкласи на PizzaBase. Дефинирајте ги овие две класи како дисјунктни.



- Креирајте ги подкласите MeatTopping, VegetableTopping, CheeseTopping и SeafoodTopping како подкласи на PizzaTopping. Дефинирајте ги овие класи како дисјунктни.

Annotations Usage
Annotations: SeafoodTopping
Annotations +

Description: SeafoodTopping
Equivalent To
SubClass Of +
PizzaTopping
General class axioms +
SubClass Of (Anonymous Ancestor)
Instances
Target for Key +
Disjoint With +
MeatTopping, CheeseTopping, VegetableTopping

5. Во рамките на класата MeatTopping додадете ги следниве дисјунктни подекласи: SpicyBeefTopping, PepperoniTopping, SalamiTopping и HamTopping.

Annotation properties Datatypes Individuals
Classes Object properties Data properties
Class hierarchy: HamTopping
Annotations Usage Annotations: HamTopping
Annotations +

Description: HamTopping
SubClass Of +
MeatTopping
General class axioms +
SubClass Of (Anonymous Ancestor)
Instances
Target for Key +
Disjoint With +
SpicyBeefTopping, PepperoniTopping, SalamiTopping
Disjoint Union Of +

6. Во рамките на класата VegetableTopping додадете ги следниве дисјунктни подекласи: TomatoTopping, OliveTopping, MushroomTopping, PepperTopping, OnionTopping и CaperTopping.

File Edit View Reasoner Tools Refactor Window Help
< > pizza (<http://www.pizza.com/ontologies/pizza.owl>) : [C:\Users\dimitr\OneDrive\Documents\pizza.owl]
PizzaTopping / VegetableTopping / CaperTopping
Active ontology Entities Individuals by class DL Query
Annotation properties Datatypes Individuals
Classes Object properties Data properties
Class hierarchy: CaperTopping
Annotations Usage Annotations: CaperTopping
Annotations +

Description: CaperTopping
Equivalent To
SubClass Of +
VegetableTopping
General class axioms +
SubClass Of (Anonymous Ancestor)
Instances
Target for Key +
Disjoint With +
MushroomTopping, OliveTopping, TomatoTopping, OnionTopping, PepperTopping

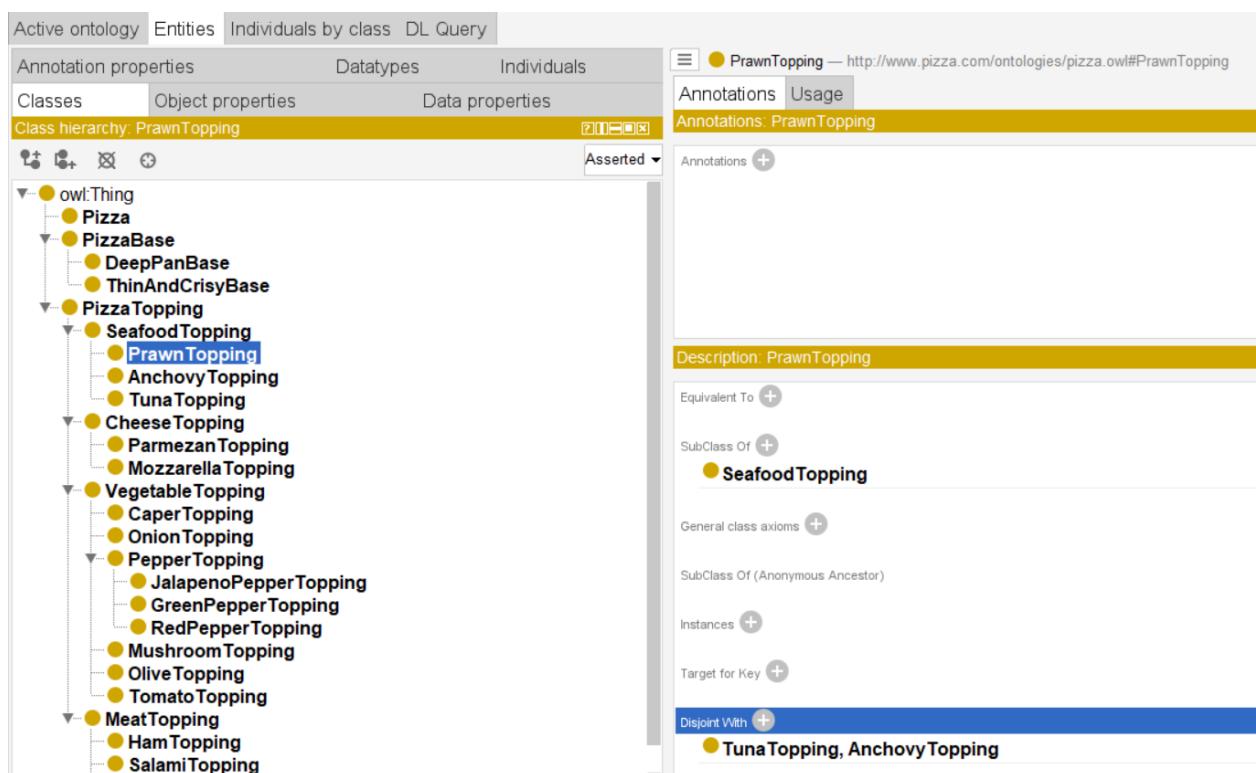
7. Во рамките на класата PepperTopping додадете ги следниве дисјунктни подкласи: RedPepperTopping, GreenPepperTopping и JalapenoPepperTopping.

The screenshot shows the Protege ontology editor interface. On the left, the class hierarchy tree includes Owl:Thing, Pizza, PizzaBase (DeepPanBase, ThinAndCrisyBase), PizzaTopping (SeafoodTopping, CheeseTopping, VegetableTopping, PepperTopping, JalapenoPepperTopping, GreenPepperTopping, RedPepperTopping, MushroomTopping, OliveTopping, TomatoTopping), MeatTopping (HamTopping, SalamiTopping, PepperoniTopping, SpicyBeefTopping). The JalapenoPepperTopping node is selected. On the right, the properties panel shows annotations for JalapenoPepperTopping, specifically defining it as a SubClass Of PepperTopping and Disjoint With RedPepperTopping, GreenPepperTopping.

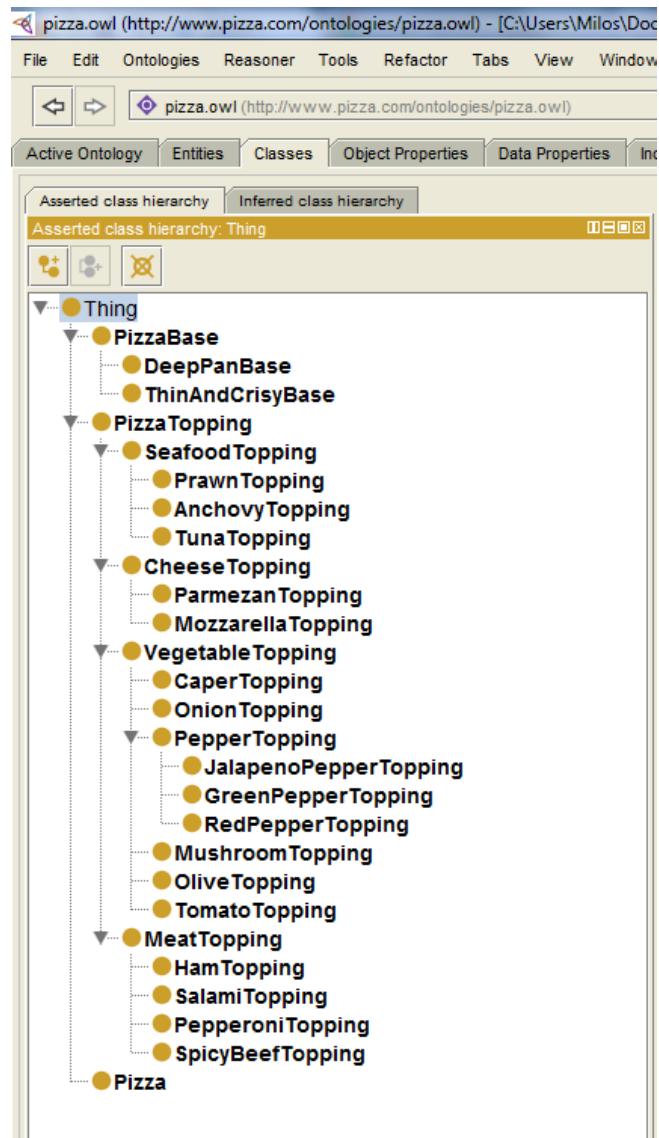
8. Во рамките на класата CheeseTopping додадете ги следниве дисјунктни подкласи: MozzarellaTopping и ParmezanTopping.

The screenshot shows the Protege ontology editor interface. On the left, the class hierarchy tree includes Owl:Thing, Pizza, PizzaBase (DeepPanBase, ThinAndCrisyBase), PizzaTopping (SeafoodTopping, CheeseTopping, VegetableTopping, PepperTopping, JalapenoPepperTopping, GreenPepperTopping, RedPepperTopping, MushroomTopping, OliveTopping, TomatoTopping), MeatTopping (HamTopping, SalamiTopping, PepperoniTopping, SpicyBeefTopping). The ParmezanTopping node is selected. On the right, the properties panel shows annotations for ParmezanTopping, specifically defining it as a SubClass Of CheeseTopping and Disjoint With MozzarellaTopping.

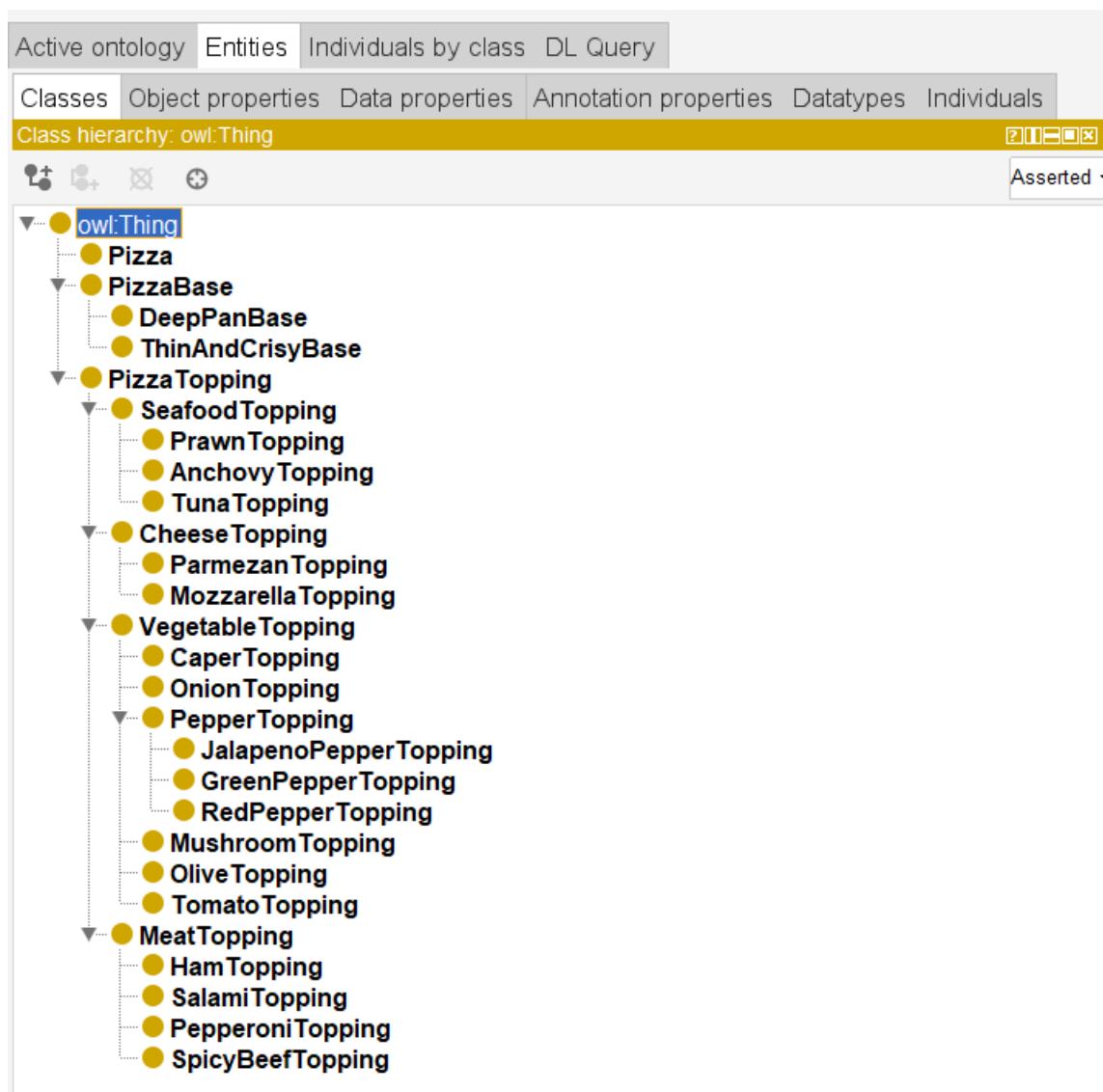
9. Во рамките на класата SeafoodTopping додадете ги следниве дисјунктни подкласи: TunaTopping, AnchovyTopping и PrawnTopping.



10. После овие точки, вашата онтологија треба да изгледа како на Слика 2. Доколку имате грешки, вратете се назад на соодветниот чекор и отстранете ги.

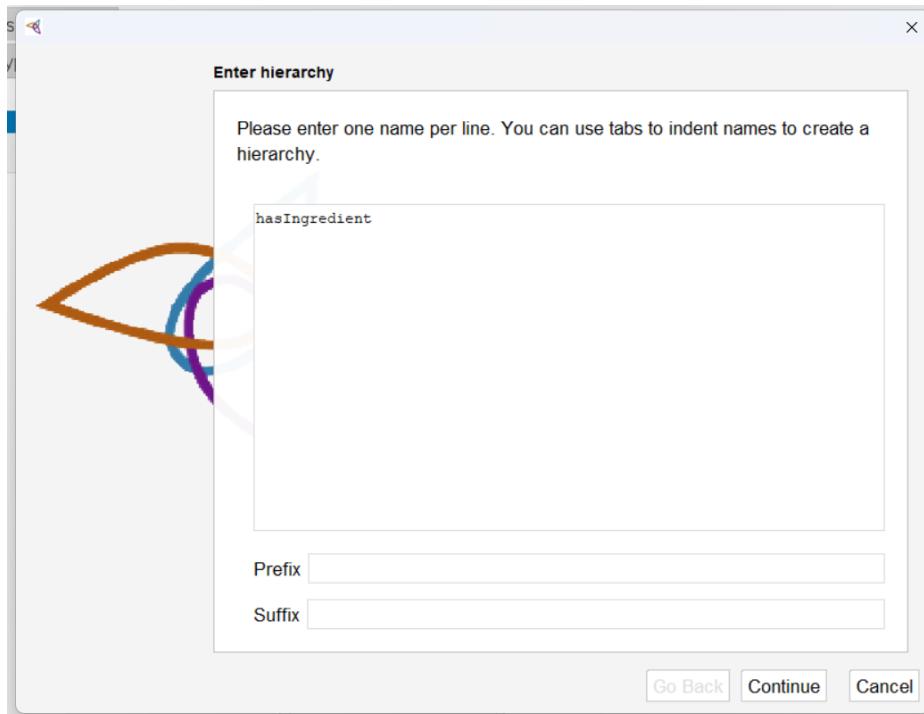


Слика 2

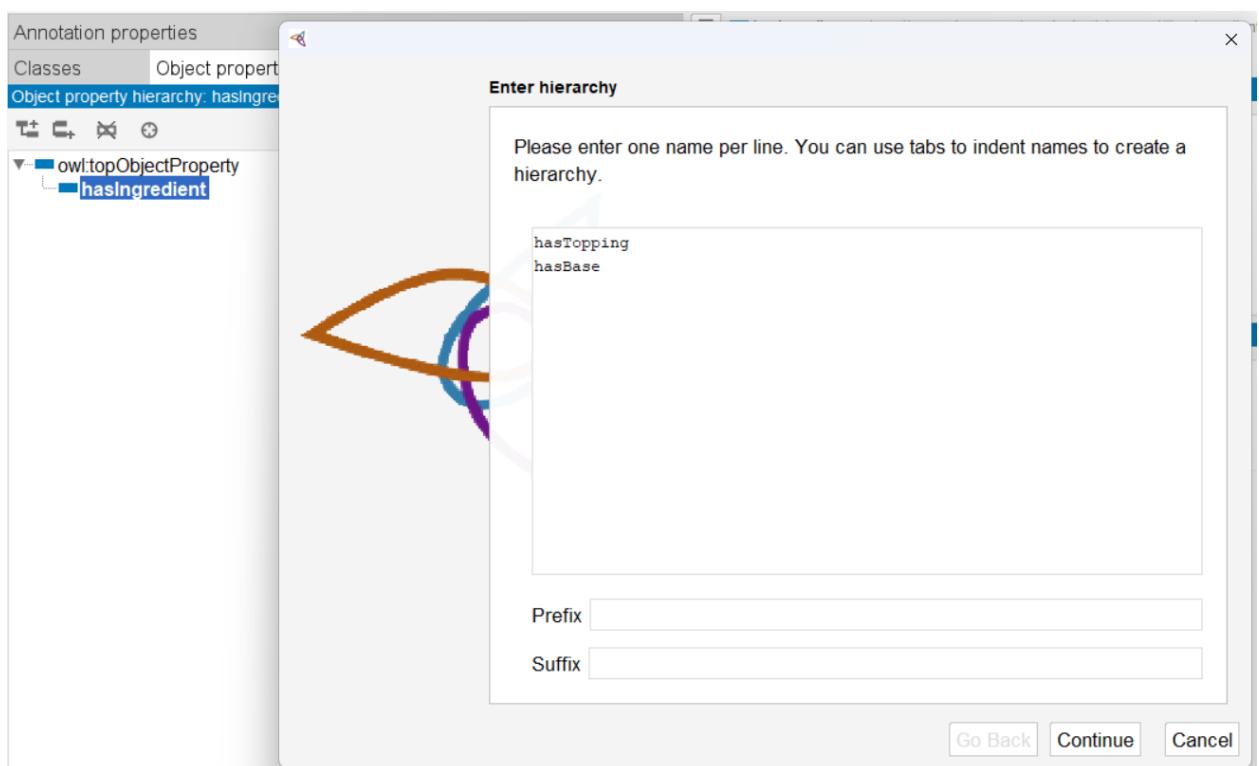


II. Дефинирање на релациите (својствата) во онтологијата

11. Префрлете се во табот Object Properties. Кликнете на копчето Add Property и додадете објектно својство со име `hasIngredient`.



12. Селектирајте го својството hasIngredient. Додадете му ги подсвојствата hasTopping и hasBase.



The screenshot shows the Protege interface with the following navigation tabs: Active ontology, Entities, Individuals by class, DL Query. Under Entities, it shows Annotation properties, Datatypes, and Individuals. Under Individuals, it shows Classes, Object properties, and Data properties. The main area displays the "Object property hierarchy: hasIngredient". The hierarchy tree shows:
 - owl:topObjectProperty
 - hasIngredient
 - hasTopping
 - hasBase

13. Креирајте ново објектно свойство, isIngredientOf. Во десниот дел од интерфејсот додадете информација дека ова свойство 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.

The screenshot shows the Protege interface with the following navigation tabs: Active ontology, Entities, Individuals by class, DL Query. Under Entities, it shows Annotation properties, Datatypes, and Individuals. Under Individuals, it shows Classes, Object properties, and Data properties. The main area displays the "Object property hierarchy: isIngredientOf". The hierarchy tree shows:
 - owl:topObjectProperty
 - hasIngredient
 - hasBase
 - hasTopping
 - isIngredientOf
 - isToppingOf
 - isBaseOf

On the right side, the "Annotations" tab is selected for the isIngredientOf property. The "Annotations: isIngredientOf" section contains:
 - Characteristics: isIngred (checkboxes for Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, Irreflexive)
 - Description: isIngredientOf (links to Equivalent To, SubProperty Of, Inverse Of, Domains (intersection), Ranges (intersection), Disjoint With, SuperProperty Of (Chain))
 - Annotations: isIngredientOf (button for adding annotations)

Annotation properties Datatypes Individuals
Classes Object properties Data properties
Object property hierarchy: hasIngredient

Annotations Usage
Annotations: hasIngredient

Characteristics: hasIngredient Description: hasIngredient

Functional Equivalent To
 Inverse functional SubProperty Of
 Transitive **owl:topObjectProperty**
 Symmetric Inverse Of
 Asymmetric **isIngredientOf**
 Reflexive Domains (intersection)
 Irreflexive Ranges (intersection)
 Disjoint With
 SuperProperty Of (Chain)

14. Креирајте ги својствата isBaseOf и isToppingOf како подсвојства на isIngredientOf.
Дефинирајте ги како инверзни својства на hasBase и hasTopping, соодветно.

Annotation properties Datatypes Individuals
Classes Object properties Data properties
Object property hierarchy: isToppingOf

Annotations Usage
Annotations: isToppingOf

Characteristics: isToppingOf Description: isToppingOf

Functional Equivalent To
 Inverse functional SubProperty Of
 Transitive **isIngredientOf**
 Symmetric Inverse Of
 Asymmetric **hasTopping**
 Reflexive Domains (intersection)
 Irreflexive

The screenshot shows the Protege interface with the 'Annotations' tab selected for the 'isBaseOf' object property. The left pane displays the 'Object property hierarchy' for 'isBaseOf', which includes 'owl:topObjectProperty', 'hasIngredient', 'hasBase', 'hasTopping', 'isIngredientOf', 'isToppingOf', and 'isBaseOf'. The right pane shows the 'Characteristics' and 'Description' tabs for 'isBaseOf'. Under 'Characteristics', the 'Transitive' checkbox is selected. Other options include 'Functional', 'Inverse functional', 'Symmetric', 'Asymmetric', 'Reflexive', and 'Irreflexive'. The 'Description' tab contains links to 'Equivalent To', 'SubProperty Of', 'Inverse Of', and 'Domains (intersection)'.

15. Селектирајте го објектното свойство hasIngredient. Дефинирајте го како транзитивно свойство, преку селектирање на check box-от даден во Characteristics (во средина на интерфејсот).

The screenshot shows the Protege interface with the 'Annotations' tab selected for the 'hasIngredient' object property. The left pane displays the 'Object property hierarchy' for 'hasIngredient', which includes 'owl:topObjectProperty', 'hasIngredient', 'hasBase', 'hasTopping', 'isIngredientOf', 'isToppingOf', and 'isBaseOf'. The right pane shows the 'Characteristics' and 'Description' tabs for 'hasIngredient'. Under 'Characteristics', the 'Transitive' checkbox is selected. Other options include 'Functional', 'Inverse functional', 'Symmetric', 'Asymmetric', 'Reflexive', and 'Irreflexive'. The 'Description' tab contains links to 'Equivalent To', 'SubProperty Of', 'Inverse Of', 'Domains (intersection)', and 'Ranges (intersection)'.

16. Дефинирајте го и својството isIngredientOf како транзитивно.

Object property hierarchy: isIngredientOf

Annotations: isIngredientOf

Characteristics: isIngredientOf

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

17. Дефинирајте го својството hasBase како функционално свойство. Што значи одредено свойство да биде функционално?

Функција е множество од подредени парови од $A \times B$, каде за секое a од A , постои единствено b од B така што $f(a)=b$.

Object property hierarchy: hasBase

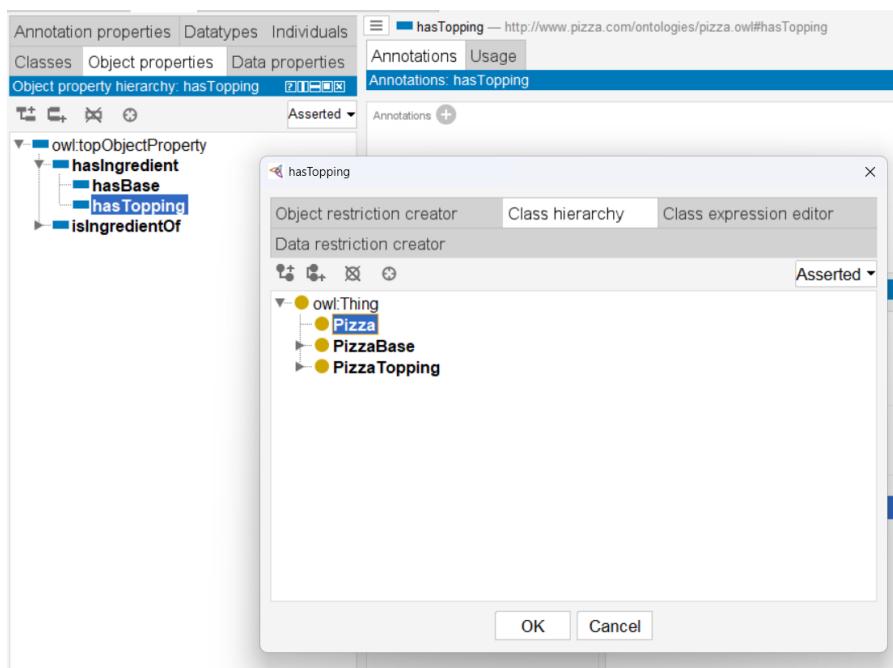
Annotations: hasBase

Characteristics: hasBase

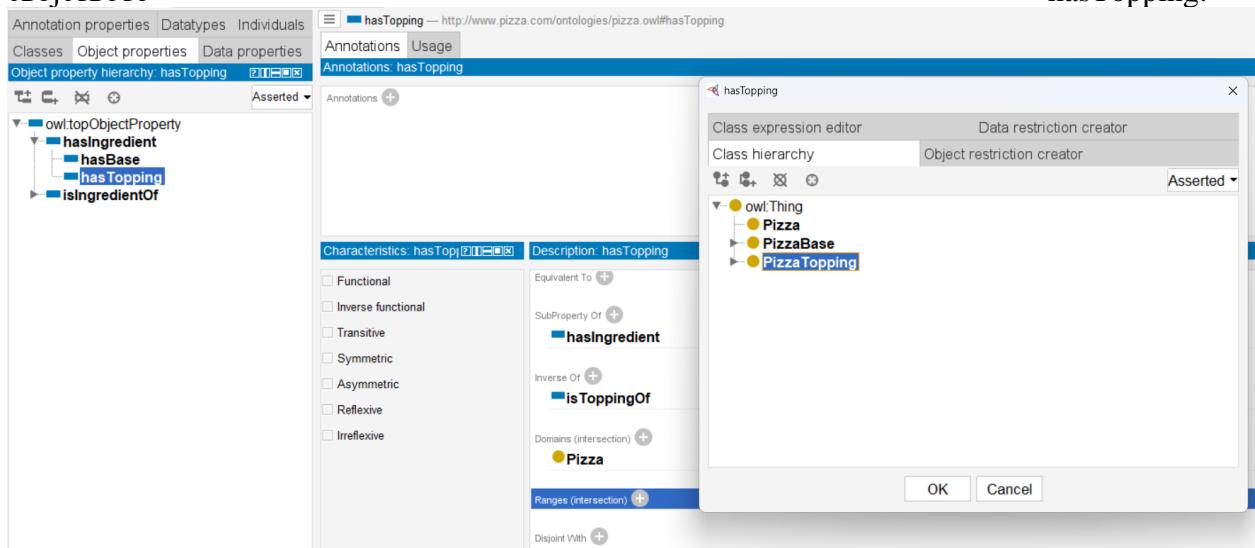
- Functional
- Inverse functional
- Transitive
- Symmetric
- Asymmetric
- Reflexive
- Irreflexive

18. Специфицирајте ги доменот и опсегот на својството hasTopping. Доменот ги одредуваше класите кои може да ги имаат овие релации, а опсегот ги одредување 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 свойството hasTopping.



Ова означува дека свойството hasTopping се однесува на инстанци од класата Pizza (доменот), а како вредност може да има инстанци од класата PizzaTopping (опсегот).

19. Специфицирајте ги доменот и опсегот на свойството 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.

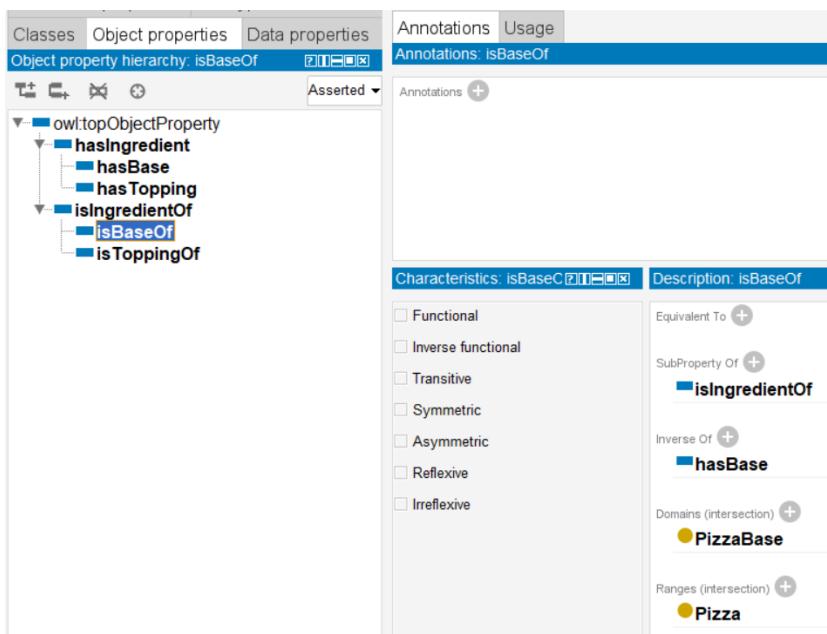
The screenshot shows the Protégé interface with the following details:

- Object property hierarchy: isToppingOf**: A tree view showing properties under owl:topObjectProperty, including hasIngredient, hasBase, hasTopping, isIngredientOf, isBaseOf, and isToppingOf.
- Annotations: isToppingOf**: A panel showing the annotation for isToppingOf. It includes a "Characteristics" section with checkboxes for Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, and Irreflexive. The "Description" section lists:
 - Equivalent To: + (empty)
 - SubProperty Of: + (empty)
 - Inverse Of: + (empty)
 - Domains (intersection): + (empty)
 - PizzaTopping
 - Ranges (intersection): + (empty)
 - Pizza

20. Специфицирајте ги доменот и опсегот на својството hasBase и на неговото инверзно својство isBaseOf. Домен на hasBase е Pizza, а опсег е PizzaBase. Каде isBaseOf доменот и опсегот се обратни, односно PizzaBase е доменот, а Pizza е опсегот.

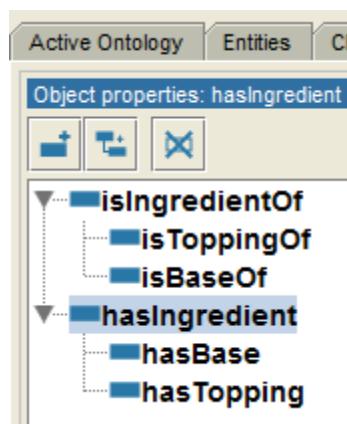
The screenshot shows the Protégé interface with the following details:

- Object property hierarchy: hasBase**: A tree view showing properties under owl:topObjectProperty, including hasIngredient, hasBase, hasTopping, isIngredientOf, isBaseOf, and isToppingOf.
- Annotations: hasBase**: A panel showing the annotation for hasBase. It includes a "Characteristics" section with checkboxes for Functional (checked), Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, and Irreflexive. The "Description" section lists:
 - Equivalent To: + (empty)
 - SubProperty Of: + (empty)
 - Inverse Of: + (empty)
 - Domains (intersection): + (empty)
 - Pizza
 - Ranges (intersection): + (empty)
 - PizzaBase

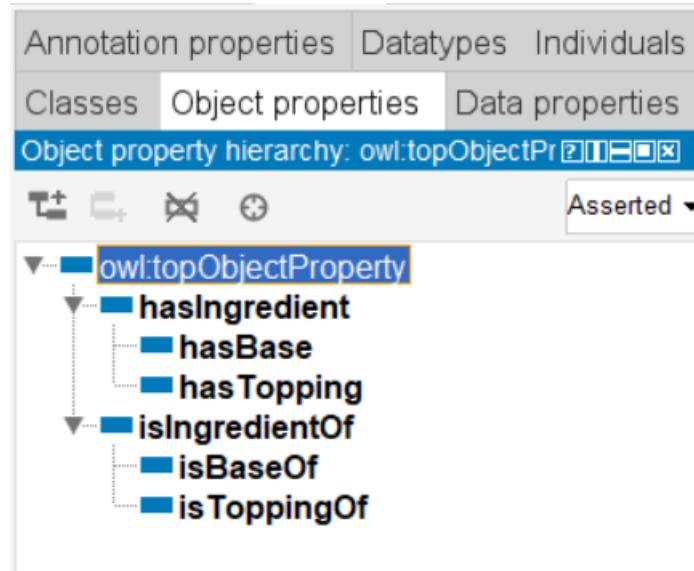


Слика 3

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



Слика 3

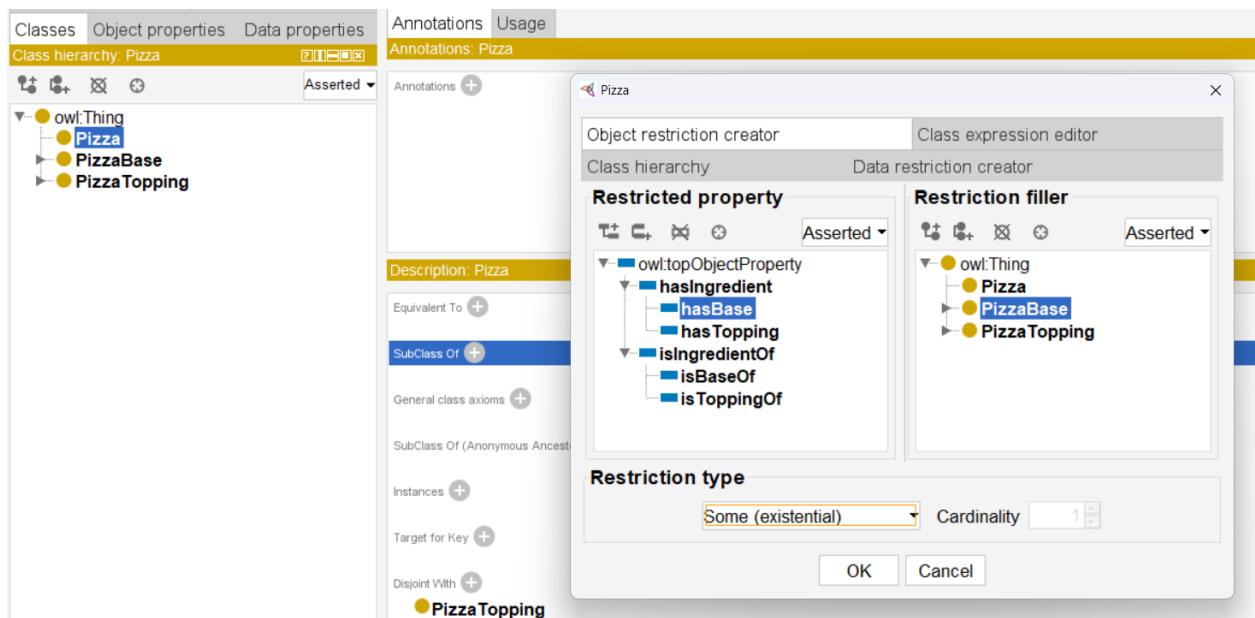


III. Дефинирање ограничувања во онтологијата

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

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 супереклса 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 OK.

На овој начин, со помош на анонимна супереклса дефиниравте ограничувања за класата Pizza.



23. Во класата Pizza дефинирајте подкласа NamedPizza. Во NamedPizza дефинирајте подкласа MargheritaPizza. Додајте го следниот коментар кај MargheritaPizza: „A pizza

that only has Mozarella and Tomato toppings“. Додавањето на коментар се прави со својството comment, во делот Annotations +.

24. Каква вредност за „Inferred anonymous superclasses“ има класата MargheritaPizza?
Зошто?

Се подразбира дека (inferred) има anonymous superclass т.е. “SubClass” – hasBase some PizzaBase. Ова го има наследено од NamedPizza, а NamedPizza го има наследено од Pizza класта.

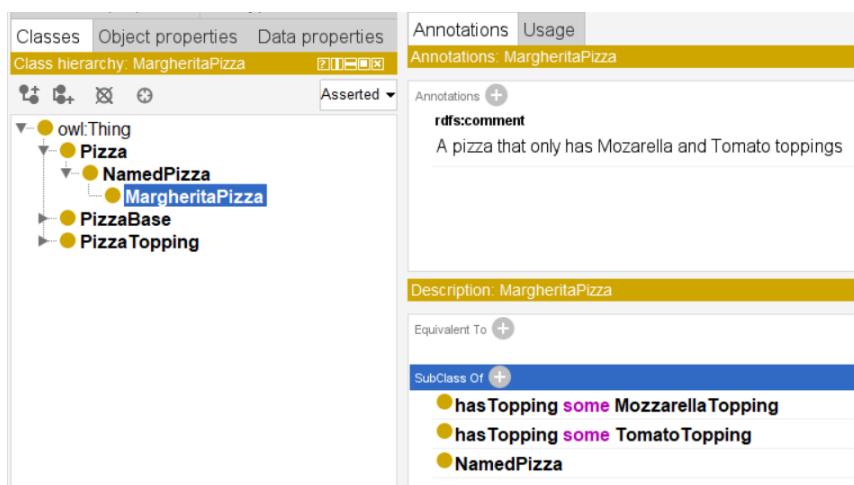
25. На сличен начин како кај точка 22, дефинирајте ограничувања за MargheritaPizza дека својството hasTopping мора да има someValueFrom од MozzarellaTopping. Каков запис има во полето Superclasses сега за MargheritaPizza?

The screenshot shows the Protégé ontology editor interface. On the left, the 'Class hierarchy' tab is selected, showing the class MargheritaPizza is a subclass of NamedPizza, which is a subclass of Pizza. The 'Annotations' tab is also selected, displaying the annotation rdfs:comment: "A pizza that only has Mozzarella and Tomato toppings". In the center, a modal dialog for creating an 'Object restriction creator' is open. The 'Restricted property' section shows 'hasIngredient' as the top-level property, with 'hasBase' and 'hasTopping' as sub-properties. The 'Restriction filler' section lists various topping classes: CheeseTopping, MozzarellaTopping, ParmesanTopping, MeatTopping, SeafoodTopping, and VegetableTopping. The 'Restriction type' section indicates 'Some (existential)' and 'Cardinality 1'. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

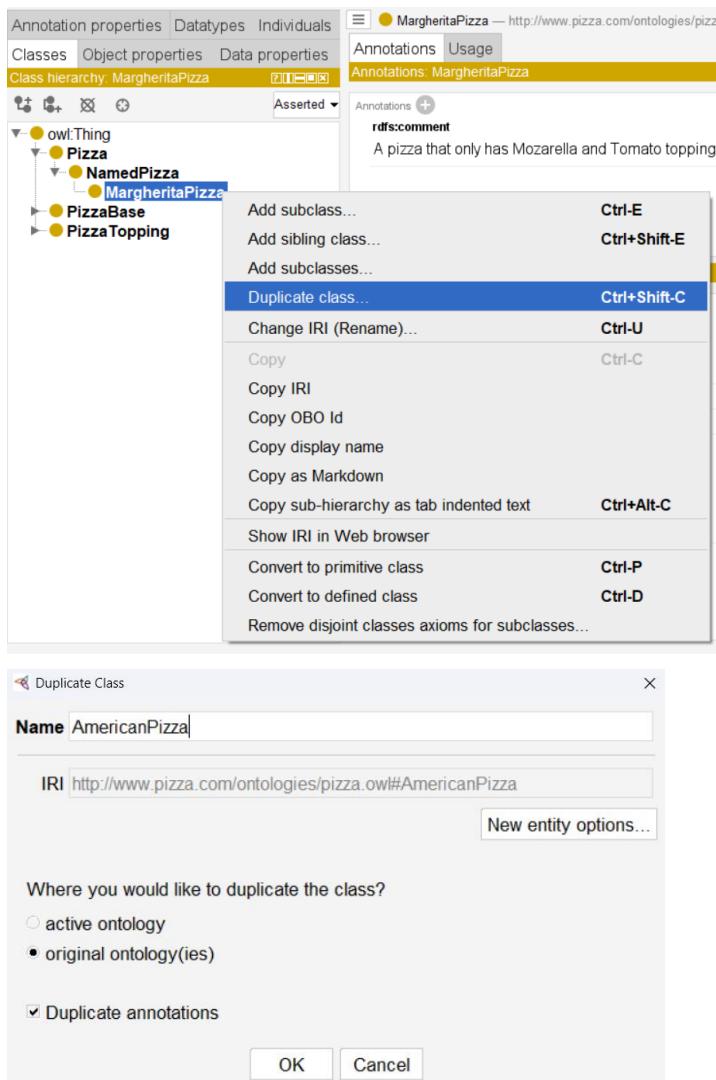
Во полето SubClass Of (Superclasses), сега го имаме новото ограничување (анонимната класа), како и фактот што наследува од NamedPizza.

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

This screenshot shows the same Protégé interface after refining the restriction. The 'Annotations' tab now includes the annotation rdfs:comment: "A pizza that only has Mozzarella and Tomato toppings". The 'Object restriction creator' dialog is open again, showing the same structure as before, but the 'Restriction filler' section now includes the specific TomatoTopping class under the VegetableTopping category. The 'OK' button is visible at the bottom of the dialog.



26. 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”. Кадо AmericanPizza дефинирајте уште едно дополнително ограничување: додадете PepperoniTopping.



The screenshot shows the Protégé ontology editor interface. At the top, there are two tabs: "Object restriction creator" and "Class expression editor". The "Object restriction creator" tab is active.

Object restriction creator:

- Restricted property:** owl:topObjectProperty, hasIngredient, hasBase, hasTopping, isIngredientOf.
- Restriction filler:** owl:Thing, Pizza, PizzaBase, PizzaTopping, CheeseTopping, MeatTopping, HamTopping, PepperoniTopping, SalamiTopping, SpicyBeefTopping, SeafoodTopping, VegetableTopping, CaperTopping, MushroomTopping, OliveTopping.
- Restriction type:** Some (existential), Cardinality: 1.

Annotations:

- Annotations: AmericanPizza** (highlighted in yellow):
 - rdfs:comment:** A pizza that only has Mozzarella and Tomato toppings
- Description: AmericanPizza** (highlighted in yellow)
- Equivalent To** (highlighted in yellow)
- SubClass Of** (highlighted in blue):
 - has Topping some MozzarellaTopping
 - has Topping some PepperoniTopping
 - has Topping some TomatoTopping
 - NamedPizza

Class hierarchy:

- AmericanPizza (highlighted in blue)
- Pizza
 - NamedPizza
 - AmericanPizza (highlighted in blue)
 - MargheritaPizza
- PizzaBase
- PizzaTopping

27. Дефинирајте ги AmericanPizza и MargheritaPizza како дисјунктни класи.

The screenshot shows the Protege ontology editor interface with the following details:

- Class hierarchy: AmericanPizza** panel: Shows the class hierarchy under `owl:Thing`. It includes `Pizza`, `NamedPizza`, `AmericanPizza`, `MargheritaPizza`, `PizzaBase`, and `PizzaTopping`.
- Annotations: AmericanPizza** panel: Shows the annotation `rdfs:comment` with the value "A pizza that only has Mozarella and Tomat".
- Description: AmericanPizza** panel: Shows the description of `AmericanPizza` with axioms: `has Topping some TomatoTopping` and `NamedPizza`.
- General class axioms**: A button to add general class axioms.
- SubClass Of (Anonymous Ancestor)**: A section showing the inheritance path: `AmericanPizza` is a subclass of `NamedPizza`, which is a subclass of `Pizza`, which is a subclass of `owl:Thing`.
- Instances**: A button to add instances.
- Target for Key**: A button to define keys for instances.
- Disjoint With**: A section showing disjointness constraints. It lists `MargheritaPizza` as a disjoint class from `AmericanPizza`.
- Annotation properties**, **Datatypes**, and **Individuals** tabs: Standard Protege tabs for managing annotations, datatypes, and individuals.
- Annotations** tab: Shows the annotation `rdfs:comment` for `AmericanPizza`.
- Expression editor** dialog: A modal window showing the detailed class hierarchy for `AmericanPizza`, including its subclasses (`NamedPizza`, `AmericanPizza`, `MargheritaPizza`) and superclasses (`PizzaBase`, `Pizza Topping`, `CheeseTopping`, `MeatTopping`, `SeafoodTopping`, `AnchovyTopping`, `PrawnTopping`, `TunaTopping`, `VegetableTopping`).

IV. Дефинирање на детали за самата онтологија

28. 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).

Веб базирани системи 2023

Active ontology Entities Individuals by class DL Query

Ontology header:

Ontology IRI <http://www.pizza.com/ontologies/pizza.owl>
Ontology Version IRI e.g. <http://www.pizza.com/ontologies/pizza.owl/1.0.0>

Annotations +

Create Annotation

Literal Entity IRI IRI Editor Property values

Value
This is an ontology about pizzas.

Language Tag
Language Tag

Datatype
xsd:string

OK Cancel

This screenshot shows the Protégé ontology editor interface. A modal dialog titled 'Create Annotation' is open, specifically for the 'rdfs:comment' property. The 'Value' field contains the text 'This is an ontology about pizzas.' The 'Datatype' dropdown is set to 'xsd:string'. The background shows the main ontology workspace where the same text is displayed as a comment for the entire ontology. The top navigation bar includes tabs for Active ontology, Entities, Individuals by class, and DL Query. The Ontology header section shows the IRI and version information. The left sidebar has sections for Annotations, Ontology imports, and Imported ontologies.

Active ontology Entities Individuals by class DL Query

Ontology header:

Ontology IRI <http://www.pizza.com/ontologies/pizza.owl>
Ontology Version IRI e.g. <http://www.pizza.com/ontologies/pizza.owl/1.0.0>

Annotations +

rdfs:comment
This is an ontology about pizzas

Create Annotation

Literal Entity IRI IRI Editor Property values

Value
Pizza Ontology

Language Tag
Language Tag

Datatype
xsd:string

OK Cancel

This screenshot shows the Protégé ontology editor interface, similar to the one above but with different annotation content. A modal dialog titled 'Create Annotation' is open, specifically for the 'rdfs:comment' property. The 'Value' field contains the text 'Pizza Ontology'. The 'Datatype' dropdown is set to 'xsd:string'. The background shows the main ontology workspace where the same text is displayed as a comment for the entire ontology. The top navigation bar includes tabs for Active ontology, Entities, Individuals by class, and DL Query. The Ontology header section shows the IRI and version information. The left sidebar has sections for Annotations, Ontology imports, and Imported ontologies.

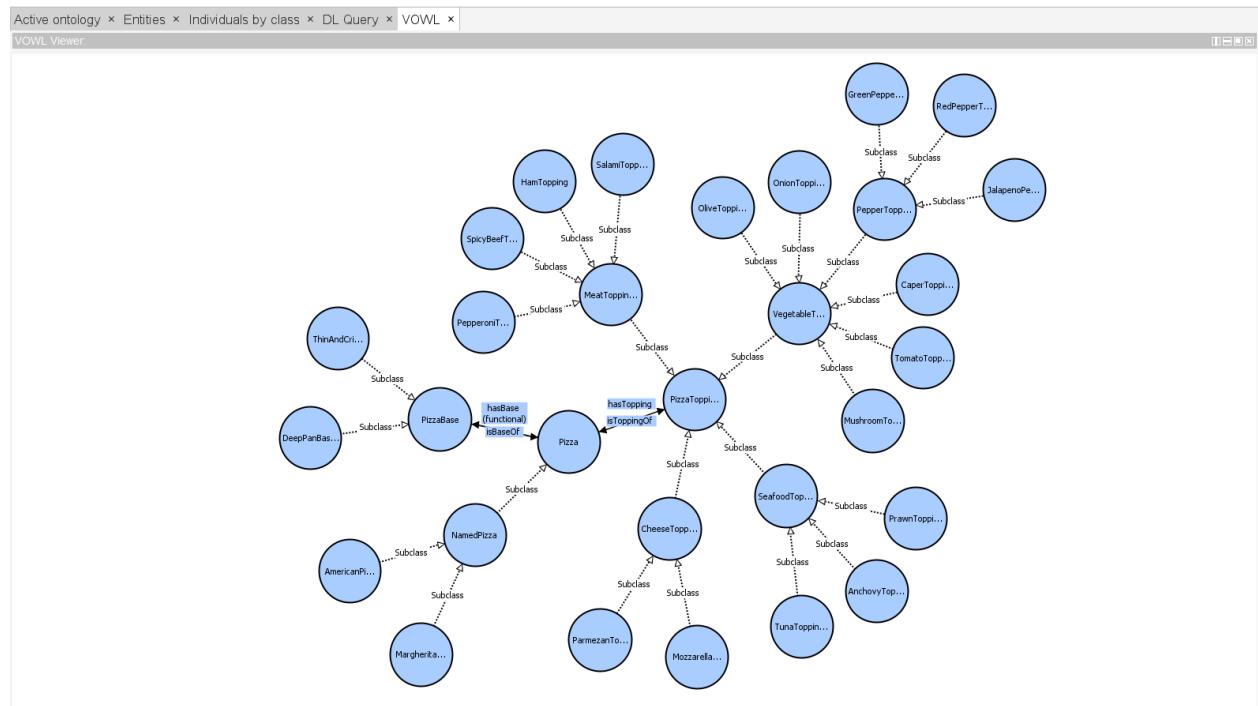
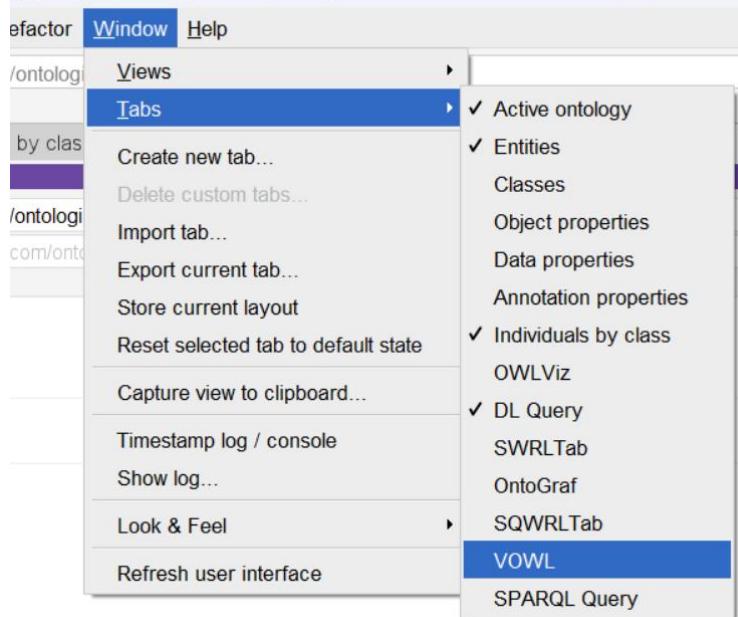
V. Визуелизација на онтологијата

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

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

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

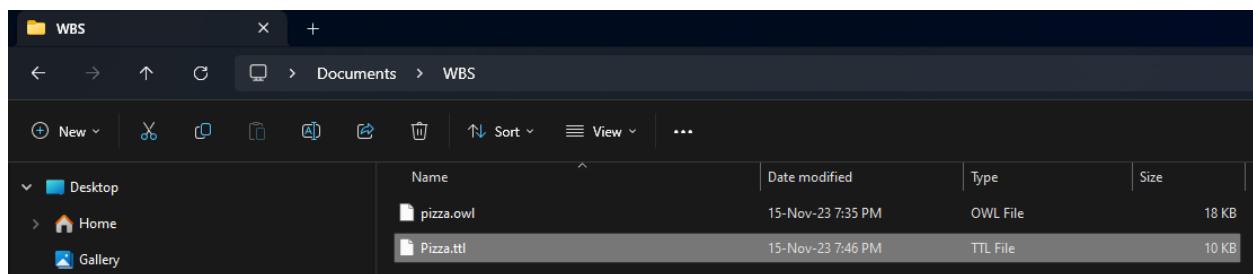
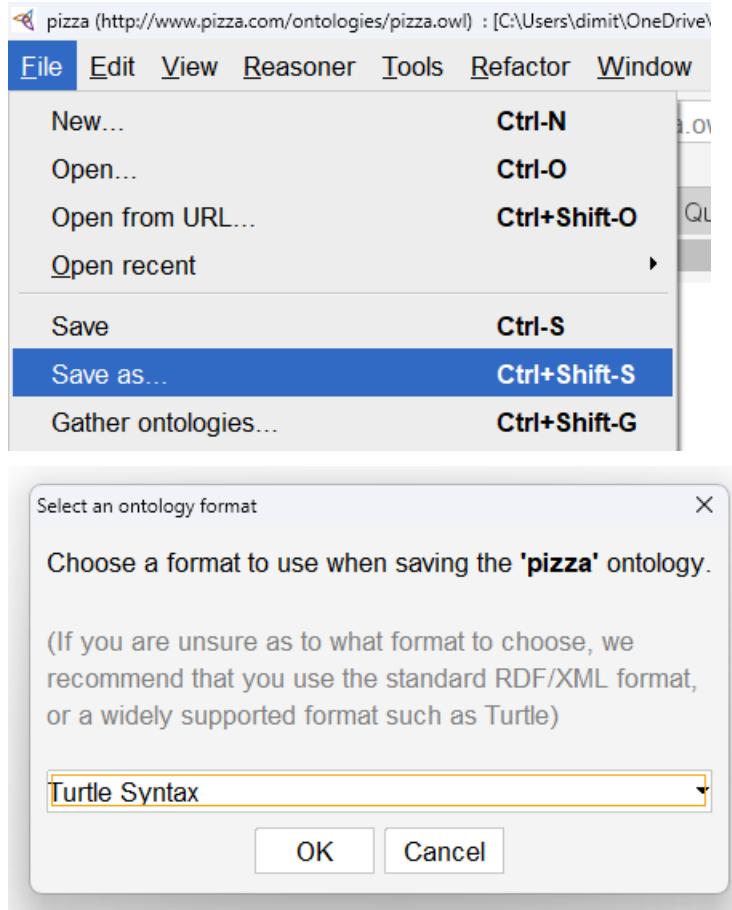
: [C:\Users\dimit\OneDrive\Documents\pizza.owl]



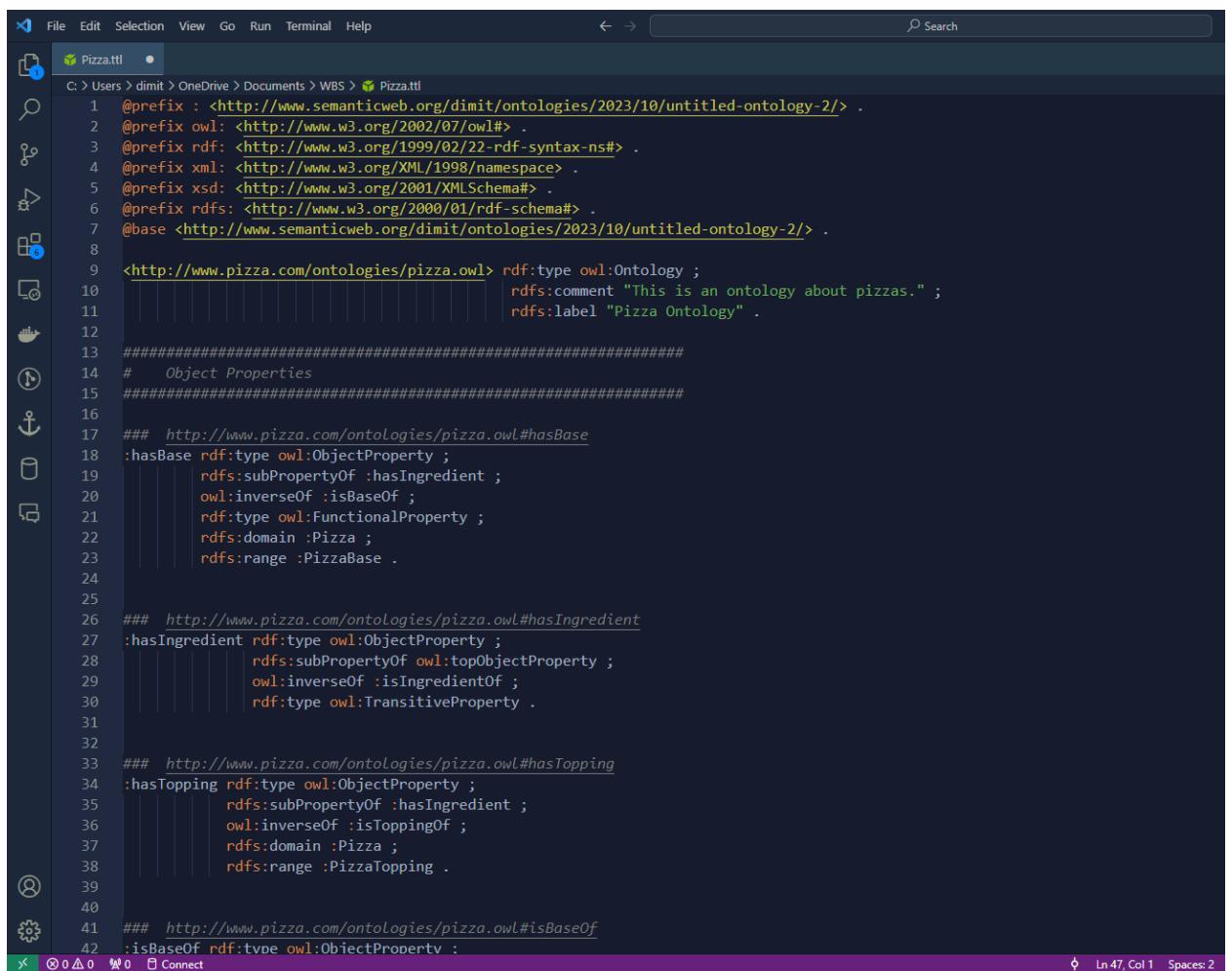
Јас ги разгледав класите и нивната поврзаност и можат да заклучам дека тоа што го очевував целосно одговара со структурата што ја добив.

VI. Онтологијата како Turtle датотека

31. 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).



32. Отворете ја Turtle датотеката во текстуален едитор и разгледајте ги креираните тројки со кои е дефинирана онтологијата од вежбата.



```
Pizza.ttl
C: > Users > dimit > OneDrive > Documents > WBS > Pizza.ttl
1 @prefix : <http://www.semanticweb.org/dimit/ontologies/2023/10/untitled-ontology-2/> .
2 @prefix owl: <http://www.w3.org/2002/07/owl#> .
3 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
4 @prefix xml: <http://www.w3.org/XML/1998/namespace> .
5 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
6 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
7 @base <http://www.semanticweb.org/dimit/ontologies/2023/10/untitled-ontology-2/> .
8
9 <http://www.pizza.com/ontologies/pizza.owl> rdf:type owl:Ontology ;
10 | rdfs:comment "This is an ontology about pizzas." ;
11 | rdfs:label "Pizza Ontology" .
12
13 ##### Object Properties #####
14 # Object Properties
15 #####
16
17 ### http://www.pizza.com/ontologies/pizza.owl#hasBase
18 :hasBase rdf:type owl:ObjectProperty ;
19 | rdfs:subPropertyOf :hasIngredient ;
20 | owl:inverseOf :isBaseOf ;
21 | rdf:type owl:FunctionalProperty ;
22 | rdfs:domain :Pizza ;
23 | rdfs:range :PizzaBase .
24
25
26 ### http://www.pizza.com/ontologies/pizza.owl#hasIngredient
27 :hasIngredient rdf:type owl:ObjectProperty ;
28 | rdfs:subPropertyOf owl:topObjectProperty ;
29 | owl:inverseOf :isIngredientOf ;
30 | rdf:type owl:TransitiveProperty .
31
32
33 ### http://www.pizza.com/ontologies/pizza.owl#hasTopping
34 :hasTopping rdf:type owl:ObjectProperty ;
35 | rdfs:subPropertyOf :hasIngredient ;
36 | owl:inverseOf :isToppingOf ;
37 | rdfs:domain :Pizza ;
38 | rdfs:range :PizzaTopping .
39
40
41 ### http://www.pizza.com/ontologies/pizza.owl#isBaseOf
42 :isBaseOf rdf:type owl:ObjectProperty :
```