

# Praktyczny przewodnik tworzenia ontologii w języku OWL za pomocą Protégé 4.2

Wykonany na podstawie

*Protégé OWL Tutorial. A step-by-step guide to modelling in OWL using the popular Protégé OWL tools.* Matthew Horridge i inni dla projektu CO-ODE.

(<http://130.88.198.11/tutorials/protegeowltutorial/>)

przez

Tomasz Dzumaga, Błażej Kącikowski, Łukasz Mocek (TSiSS 2012/2013)

i edytowany przez Agnieszkę Ławrynowicz, 28.10.2013, 2.11.2014



Ten utwór jest dostępny na [licencji Creative Commons Uznanie autorstwa-Na tych samych warunkach 4.0 Międzynarodowe](https://creativecommons.org/licenses/by-sa/4.0/).

## 1. Tworzenie ontologii w języku OWL.

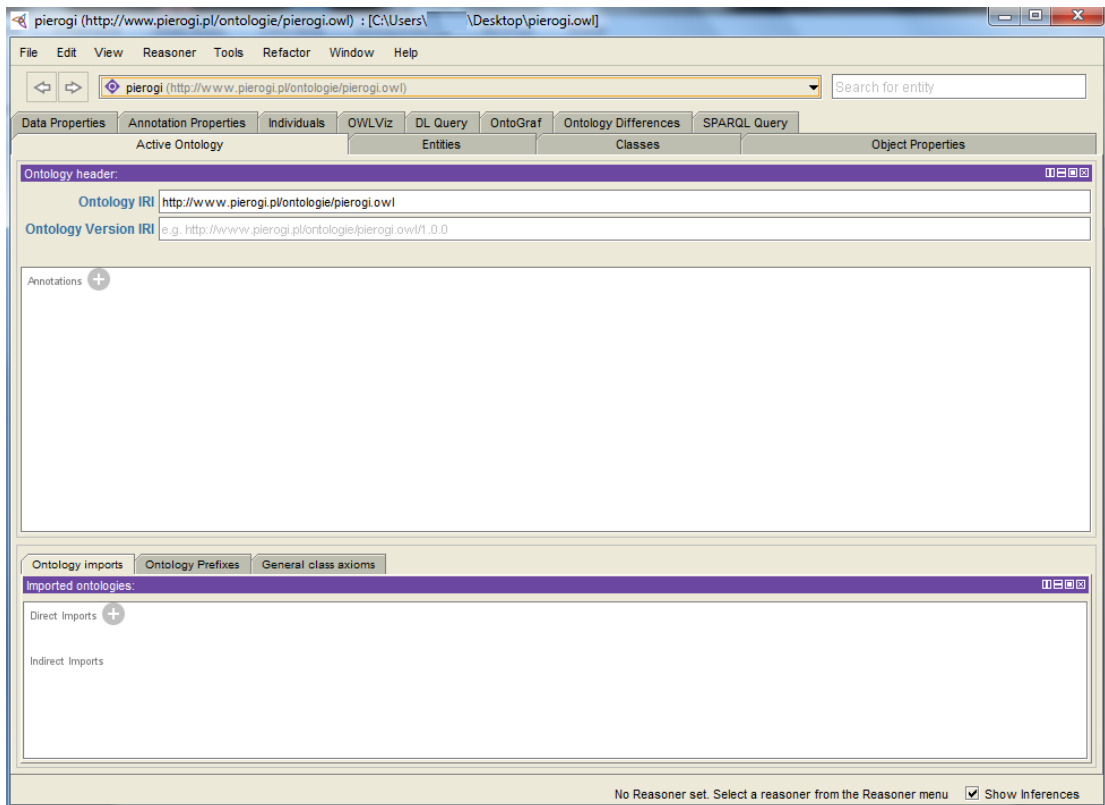
W niniejszym tutorialu będziemy tworzyć ontologię pierogów.

### Ćwiczenie 1. Tworzenie nowej ontologii.

---

1. Uruchom Protégé.
  2. Wybierz opcję tworzenia nowej ontologii. Każdej ontologii zostaje przydzielony identyfikator za pomocą *Internationalized Resource Identifier (IRI)*. Zastąp domyślnie wygenerowany IRI w polu *Ontology IRI* poprzez <http://www.pierogi.pl/ontologie/pierogi.owl>. Wybierz format *RDF/XML*.
  3. Pojawi się okno z widoczną zakładką *Active Ontology* (widoczne na rysunku 1).
  4. Żeby nie stracić wyników pracy z menu *File* wybierz *Save as...* Pojawi się okienko wyboru formatu ontologii. Wybierz format *RDF/XML* i kliknij *OK*. W następnym okienku wybierz miejsce zapisu swojego pliku (tam, gdzie posiadasz do tego prawa, czyli np. na Pulpicie), podaj nazwę pliku i kliknij *Save*.
- 

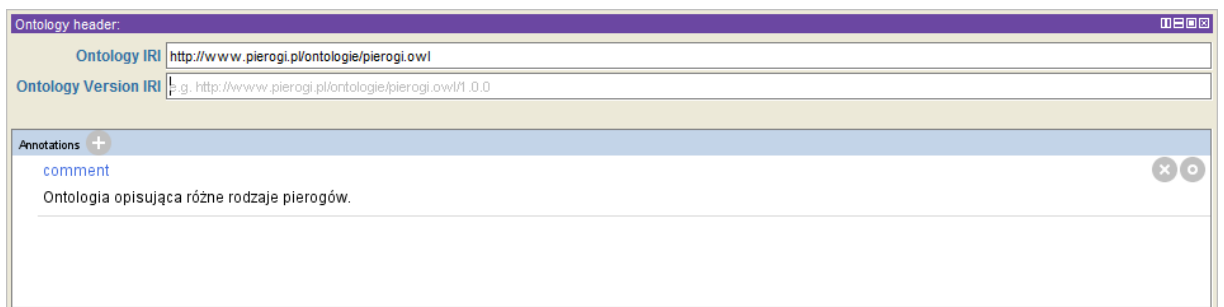
Jak widać na Rysunku 1, zakładka *Active Ontology* dostarcza informacji na temat określonej ontologii. Na przykład IRI ontologii może zostać zmieniony, adnotacje dotyczące ontologii, takie jak np. komentarze, informacje o twórcy, wersji itp. mogą być dodawane lub edytowane, także przestrzenie nazw i importy mogą być ustalone za pomocą tej zakładki.



Rysunek 1.

## Ćwiczenie 2. Dodawanie komentarza do ontologii.

1. Upewnij się, że zakładka *Active Ontology* jest aktywna.
2. W widoku *Annotations* kliknij ikonę *Add* (+) obok napisu *Annotations*. W efekcie pojawi się okno edycji. Z menu po lewej stronie wybierz *comment* i w polu z prawej strony wpisz dowolny tekst.
3. Kliknij *OK*. W tym momencie widok adnotacji powinien wyglądać podobnie jak na Rysunku 2.



Rysunek 2.

## 2. Nazwane klasy.

Ontologia zawiera klasy, są one głównymi elementami ontologii w języku OWL. W Protégé 4, edycja klas jest obsługiwana poprzez zakładkę *Classes*, przedstawioną na Rysunku 3. Początkowe drzewo hierarchii klas powinno przypominać to, które zostało przedstawione na Rysunku 4. Pusta ontologia zawiera tylko jedną klasę nazywaną *Thing*. Klasy OWL są interpretowane jako zbiory *jednostek* (lub zbiory obiektów/indywiduów). Klasa *Thing* reprezentuje zbiór zawierający wszystkie jednostki. Z tego powodu wszystkie klasy są podklasami klasy *Thing*.

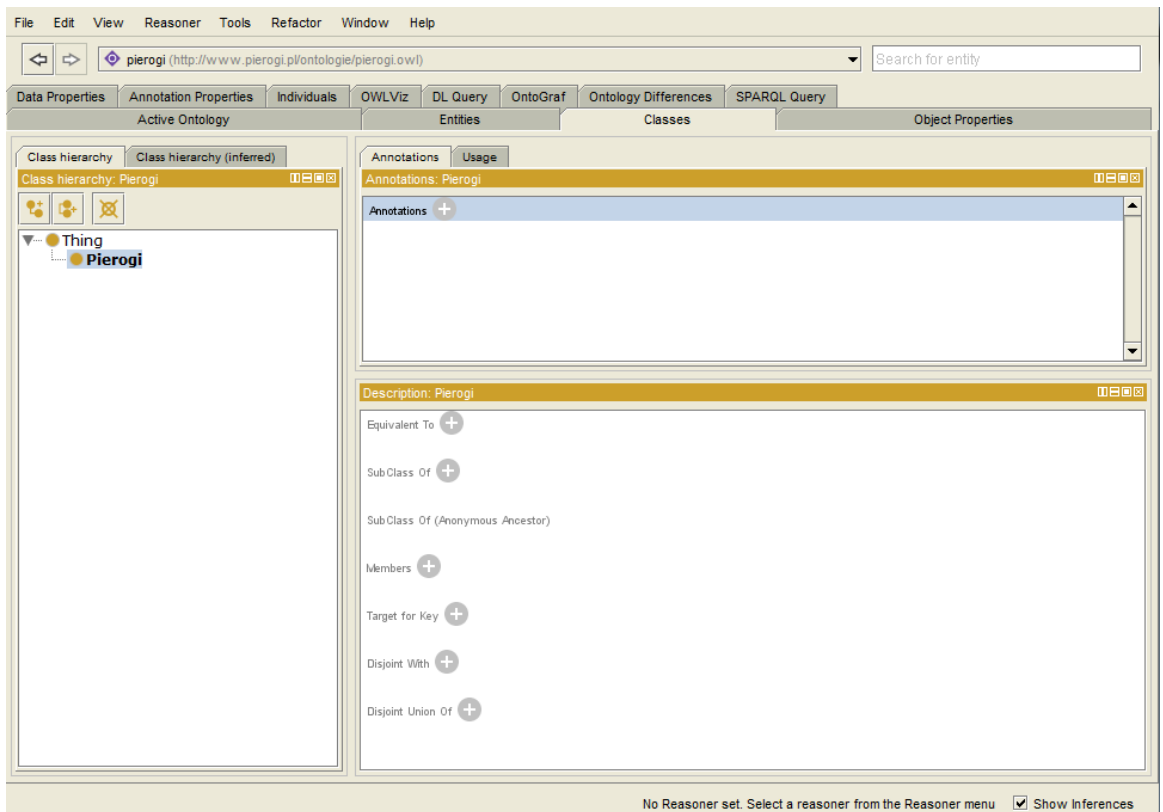
### Ćwiczenie 3. Tworzenie klas *Pierogi*, *PierogiNadzienie* i *PierogiCiasto*.

---

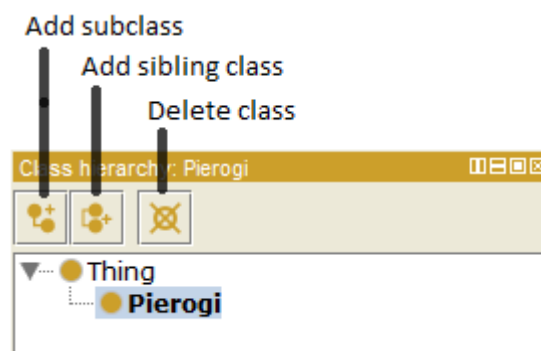
1. Upewnij się, że zakładka *Classes* jest aktywna.
2. Kliknij na klasę *Thing*, a następnie na ikonę *Add subclass*, zaznaczoną na Rysunku 4. Ten przycisk tworzy nową klasę jako podklasę zaznaczonej klasy (w tym przypadku chcemy stworzyć podklasę klasy *Thing*).
3. Pojawi się okienko, w którym należy podać nazwę klasy. Wprowadź tekst *Pierogi* (tak jak przedstawiono na rysunku 5) i naciśnij *OK*.
4. Powtórz powyższe kroki, aby dodać klasy *PierogiNadzienie* i *PierogiCiasto*, upewniając się, że klasa *Thing* jest zaznaczona, zanim klikniesz na ikonę *Add subclass*, dzięki czemu te klasy zostaną stworzone jako podklasy *Thing*.

---

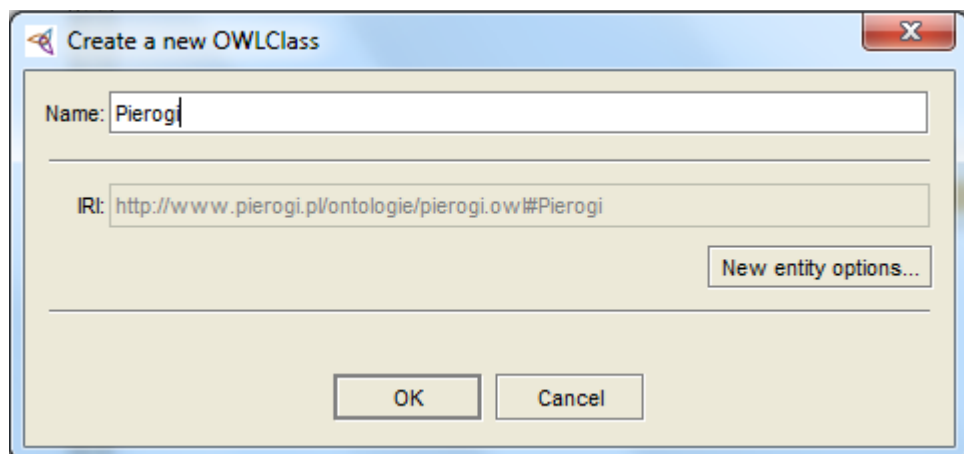
Hierarchia klas powinna przypominać hierarchię, widoczną na rysunku 6.



Rysunek 3.



Rysunek 4.



Rysunek 5.



Po stworzeniu klasy *Pierogi*, zamiast ponownie wybierać klasę *Thing* i używać przycisku *Add subclass* do stworzenia klas *PierogiNadzienie* i *PierogiCiasto* jako podklasy *Thing*, można użyć przycisku *Add sibling class* (pokazanego na rysunku 4). W tym celu należy zaznaczyć klasę *Pierogi* i kliknąć wspomniany przycisk.

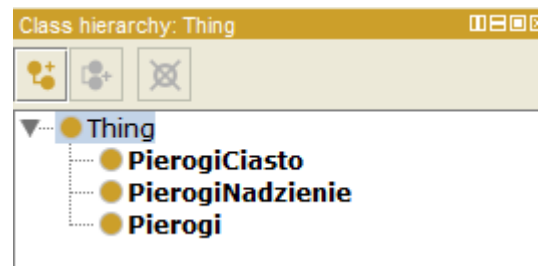
Vocabulary



Hierarchia klas może być także nazywana taksonomią.



Co prawda nie ma żadnej obowiązkowej konwencji nazywania klas OWL, jednak zalecane jest, aby nazwy klas zaczynały się od wielkiej litery i nie zawierały żadnych spacji, np. *PierogiNadzienie*. Alternatywnie można także używać podkreśleń do oddzielenia słów w nazwie np. *Pierogi\_Nadzienie*. Najważniejsze to trzymać się wybranej przez siebie konwencji.



Rysunek 6.

### 3. Klasy rozłączne.

Mając dodane klasy *Pierogi*, *PierogiNadzienie*, *PierogiCiasto* do naszej ontologii, możemy już powiedzieć, że są to klasy rozłączne, zatem każdy obiekt może być instancją tylko jednej z tych klas. Aby określić, które klasy są rozłączne należy kliknąć przycisk *Disjoint With* znajdujący się na dole widoku *Description* na zakładce *Classes*.

#### Ćwiczenie 4. Modelowanie rozłączności klas *Pierogi*, *PierogiNadzienie* i *PierogiCiasto*.

---

1. Zaznacz klasę *Pierogi* na drzewie hierarchii klas.
  2. Kliknij przycisk *Disjoint With*. Spowoduje to pojawienie się nowego okna, w którym należy zaznaczyć wszystkie klasy i kliknąć *OK*. Spowoduje to, że klasy *PierogiNadzienie* i *PierogiCiasto*, będą rozłączne z klasą *Pierogi*.
-

Zauważ, że w widoku *Description*, pod przyciskiem *Disjoint With*, pojawiły się nazwy klas *PierogiNadzienie* i *PierogiCiasto*. Po kliknięciu na klasę *PierogiNadzienie*, w tym samym miejscu wypisane będą klasy *Pierogi* i *PierogiCiasto*, które są rozłączne z klasą *PierogiNadzienie*.

#### MEANING



Klasy OWL z założenia mogą na siebie „nachodzić”. Nie możemy jednak założyć, że dany obiekt nie jest członkiem konkretnej klasy tylko dlatego, że nie został do niej jednoznacznie przypisany. Aby ‘oddzielić’ grupę klas musimy sprawić, aby były one wzajemnie rozłączne. W ten sposób można zagwarantować, że żaden obiekt należący do klasy w danej grupie, nie może także należeć do innej klasy w tej grupie. W naszym przykładzie klasy *Pierogi*, *PierogiNadzienie*, *PierogiCiasto* są wzajemnie rozłączne. Nie ma możliwości, aby dany obiekt był członkiem każdej z tych klas – nie miało by sensu, by ciasto do pierogów było jednocześnie ich nadzieniem.

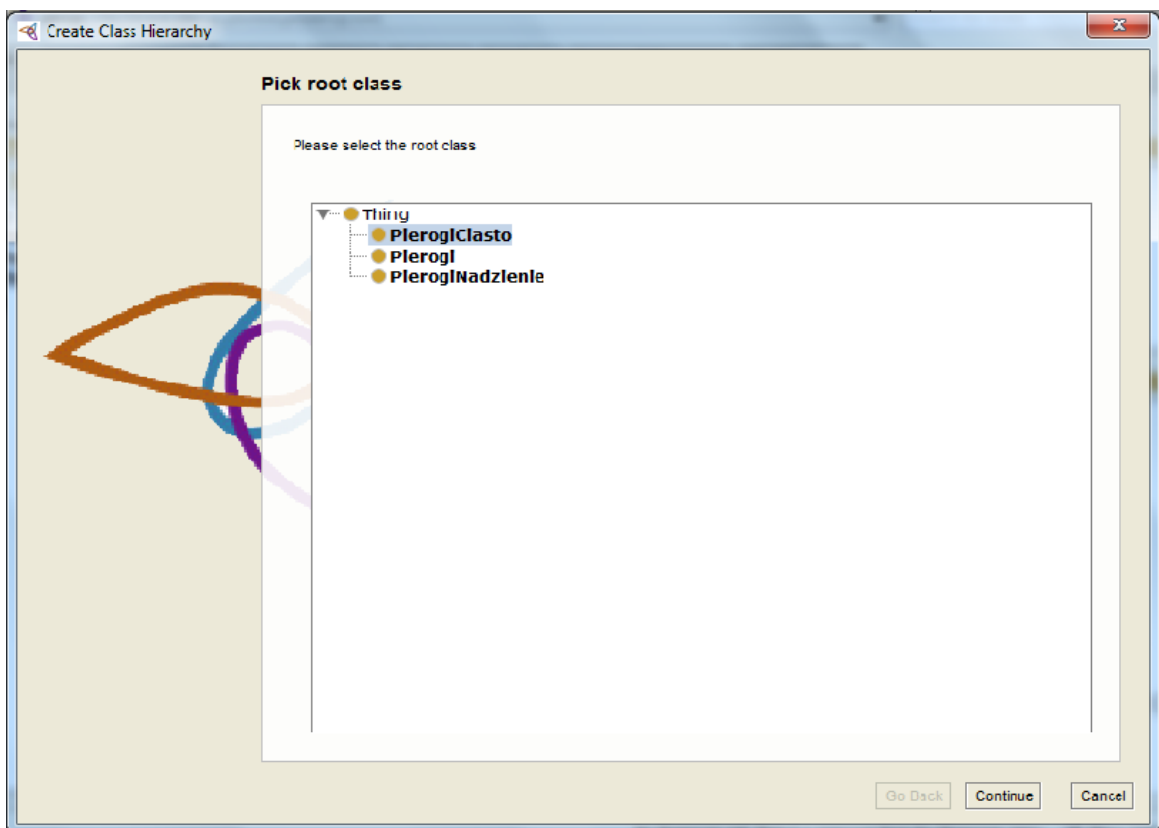
## 4. Używanie narzędzia *Create class hierarchy* do tworzenia klas.

W tym rozdziale pokażemy, jak użyć narzędzia *Create class hierarchy* do dodania kilku podklas do klasy *PierogiCiasto*.

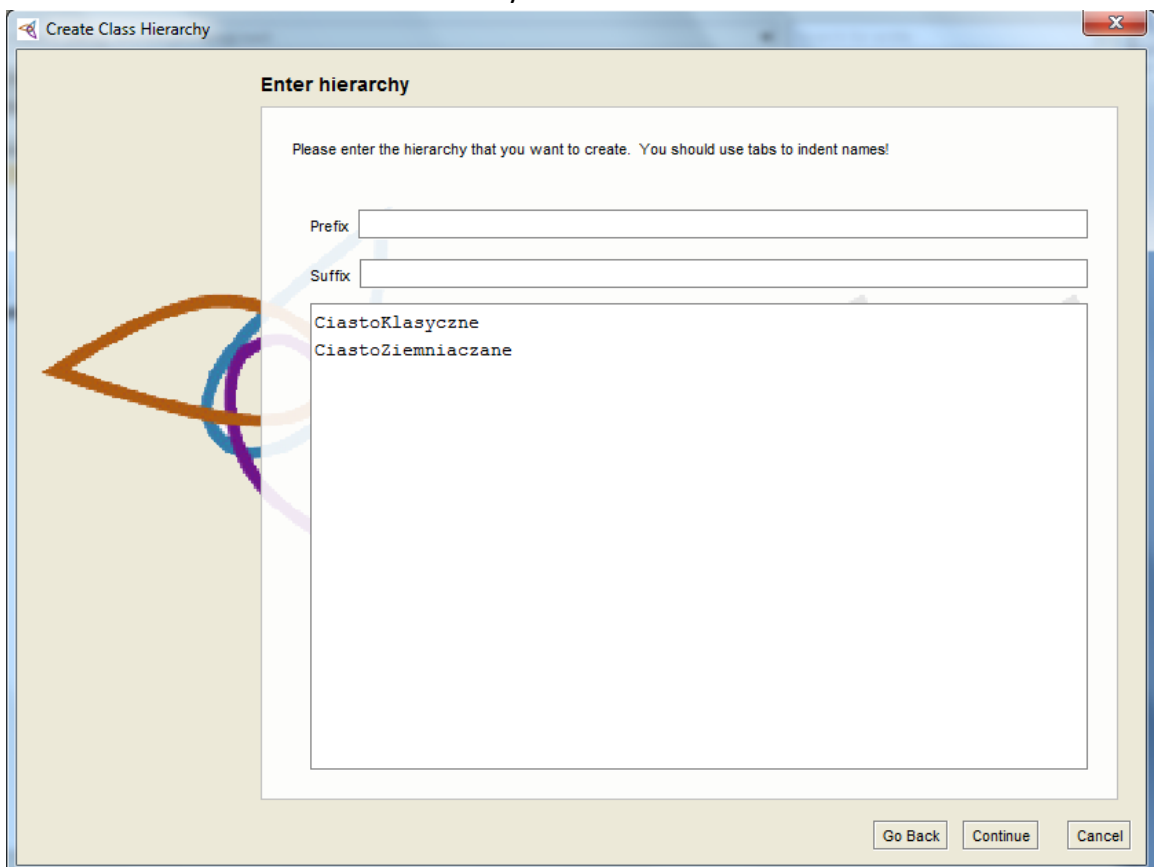
### Ćwiczenie 5. Użycie narzędzia *Create class hierarchy* do stworzenia klas *CiastoKlasyczne* i *CiastoZiemniaczane* jako podklas klasy *PierogiCiasto*.

---

1. Zaznacz klasę *PierogiCiasto* z drzewa hierarchii klas.
  2. Na pasku menu programu Protégé, wybierz *Tools*, a następnie *Create class hierarchy*.
  3. Pokaże się okno widoczne na Rysunku 7. Ponieważ zaznaczona została klasa *PierogiCiasto*, w nowo otwartym oknie powinna być ona domyślnie zaznaczona, zachęcając w ten sposób do tworzenia jej podklas. Jeżeli nie zaznaczono tej klasy tylko od razu wywołano narzędzie *Create class hierarchy*, można w tym momencie wybrać interesującą nas klasę.
  4. Naciśnij przycisk *Continue*. Pojawi się okno widoczne na Rysunku 8. Musimy poinformować narzędzie o tym jakie podklasy *PierogiCiasto* chcemy stworzyć. W dużym pustym polu wpisz *CiastoKlasyczne* i wciśnij *Enter*. Następnie wpisz *CiastoZiemniaczane*. Całość powinna wyglądać tak, jak na Rysunku 8.
  5. Naciśnij *Continue*. Protégé sprawdza, czy nazwy klas są unikalne oraz, czy nie zawierają spacji. Jeśli byłyby jakieś błędy, to w tym momencie powinien pojawić się stosowny komunikat.
  6. Upewnij się, że opcja *Make sibling classess disjoint* jest zaznaczona i naciśnij *Finish*. W ten sposób utworzone klasy będą klasami rozłącznymi i nie będzie trzeba robić tego ręcznie.
-



Rysunek 7.



Rysunek 8.



Po kliknięciu przycisku *Finish*, Protege utworzy nam rozłączne klasy. Ontologia powinna teraz zawierać *CiastoKlasyczne* i *CiastoZiemniaczane* jako podklasy *PierogiCiasto*. Zatem ciasto na pierogi nie może być jednocześnie ciastem ziemniaczanym i ciastem klasycznym.

### TIP

Na pierwszej stronie *Create class hierarchy wizard*, gdzie wprowadzane są nazwy klas, jeśli chcemy stworzyć wiele klas posiadających ten sam prefix albo suffix, możemy użyć odpowiednich opcji do automatycznego dodania odpowiednich tekstów do nazw klas.

Mając kilka podstawowych klas, możemy stworzyć klasy odpowiadające za nadzienia do pierogów. Aby były one później użyteczne, zostaną pogrupowane w różne kategorie: nadzienie mięsne, nadzienie owocowe, nadzienie serowe oraz nadzienie mieszane.

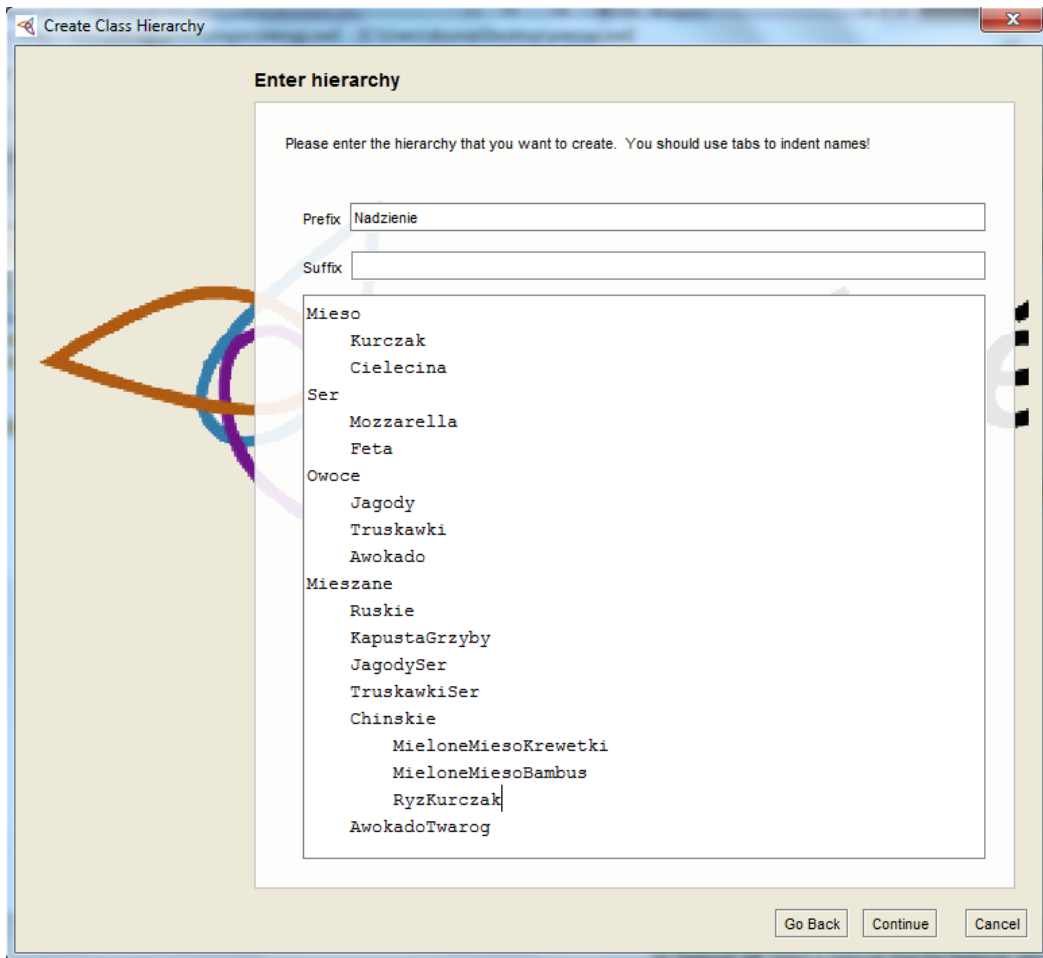


Możesz pominąć kolejne zadanie, które wymaga czasochłonnego wprowadzania danych i po prostu wczytać plik *PierogiNadzienieHierarchia.owl*, który zawiera wynik zadania.

## Ćwiczenie 6. Tworzenie podklas *PierogiNadzienie*.

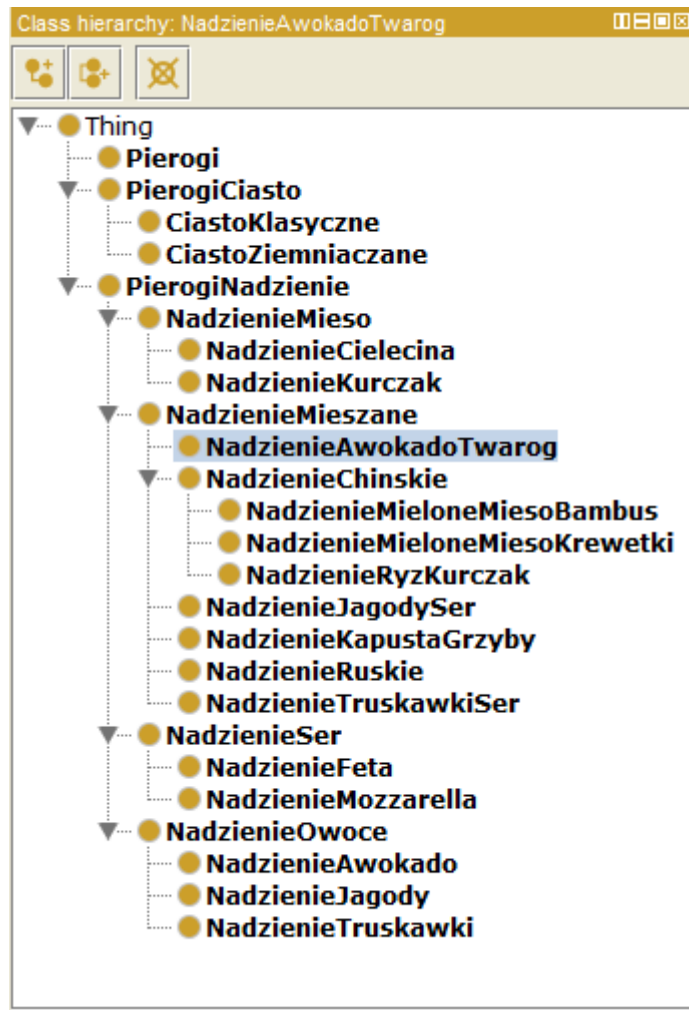
---

1. Zaznacz klasę *PierogiNadzienie* z drzewa hierarchii klas.
  2. Uruchom *Create class hierarchy* tak jak poprzednio.
  3. Upewnij się, że *PierogiNadzienie* jest zaznaczone i kliknij *Continue*.
  4. Chcemy, aby wszystkie nazwy klas zaczynały się od słowa „Nadzienie”, więc w polu *Prefix* wpisz *Nadzienie*. Dzięki temu automatycznie każda wprowadzana przez nas nazwa, będzie zaczynać się od podanego słowa.
  5. Jest możliwość już na tym etapie wprowadzenia odpowiedniej hierarchii nowych klas. Odbywa się to za pomocą tabulatora. Wprowadź nazwy klas tak jak pokazano na Rysunku 9. Zauważ, że niektóre nazwy klas muszą być poprzedzone tabulatorem, np. nazwa klasy *Kurczak*, która ma być podklasą klasy *Mieso*, została wprowadzona z użyciem przycisku *Tab*.
  6. Po wprowadzeniu wszystkich nazw klas, naciśnij *Continue*. Upewnij się, że opcja *Make sibling classess disjoint* jest zaznaczona, aby wszystkie tworzone klasy były rozłączne.
  7. Kliknij *Finish*, aby utworzyć klasy.
-



Rysunek 9.

Hierarchia klas powinna teraz wyglądać tak jak na rysunku 10.



Rysunek 10.

Oczywiście wszystkie elementy należące do klasy *NadzienieFeta*, należą także do klas *NadzienieSer* oraz *PierogiNadzienie*.

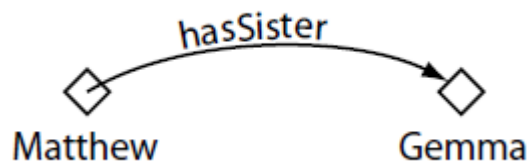
#### MEANING



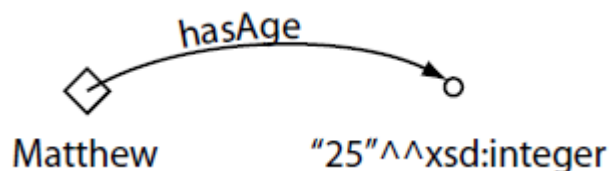
W tym momencie mamy już za sobą tworzenie kilku nazwanych klas, z których niektóre są podklasami innych klas. Konstrukcja hierarchii klas może wydawać się intuicyjna, jednakże, czy wiemy co naprawdę w OWLu oznacza, że coś jest podklasą? Np. co znaczy, że *NadzienieMieso* jest podklasą *PierogiNadzienie*. W OWLu *podklasa* oznacza *konieczną implikację*. Inaczej mówiąc, jeżeli *NadzienieMieso* jest podklasą *PierogiNadzienie*, to wszystkie instancje *NadzienieMieso* są instancjami klasy *PierogiNadzienie*. Nie ma wyjątków.

## 5. Własności OWL.

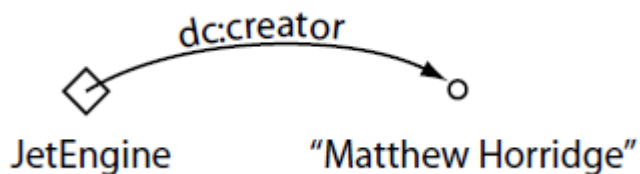
Własności OWL reprezentują relacje. Istnieją dwa główne typy własności, *obiektove* i *literałowe*. Własności obiektowe reprezentują relacje pomiędzy dwoma obiektami. W tym rozdziale skupimy się na własnościach obiektowych, własności literałowe są przedstawione w kolejnym rozdziale. Własności obiektowe łączą obiekty między sobą. OWL ma także trzeci typ własności: własności *adnotacyjne*, które mogą być używane do dodania informacji w postaci notatek do klas, obiektów i ich własności. Na Rysunku 11 przedstawione zostały przykłady każdego typu własności.



Własność obiektowa łącząca obiekt Matthew z obiektem Gemma.



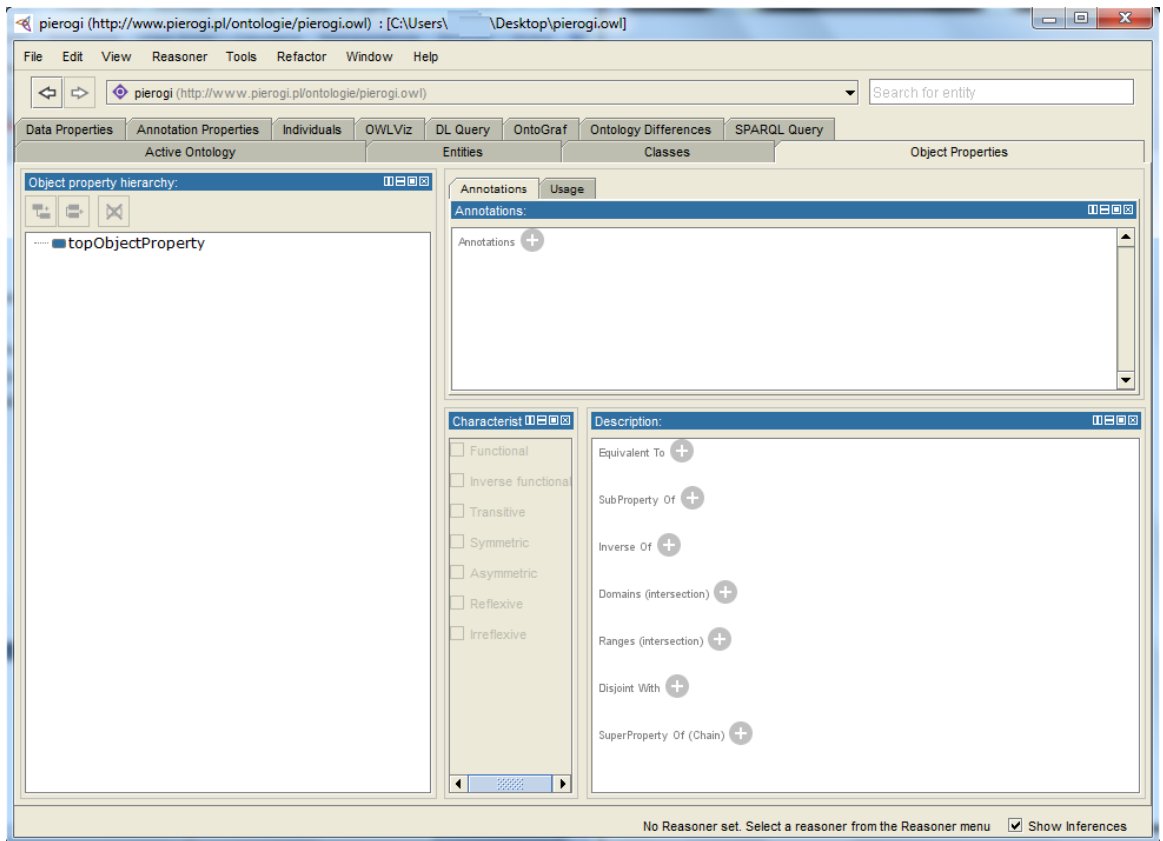
Własność literałowa łącząca obiekt Matthew z wartością „25”, będącą typu xsd:integer



Własność adnotacyjna łącząca klasę JetEngine z wartością „Matthew Horridge”

Rysunek 11.

Własności mogą być tworzone za pomocą zakładki *Object Properties* widocznej na Rysunku 12. Rysunek 13 przedstawia przyciski znajdujące się w lewym górnym narożniku zakładki *Object Properties*, które są używane do tworzenia własności OWL. Za pomocą tych przycisków można tworzyć wszystkie rodzaje własności. Większość własności tworzonych w tym tutorialu to będą własności obiektowe.



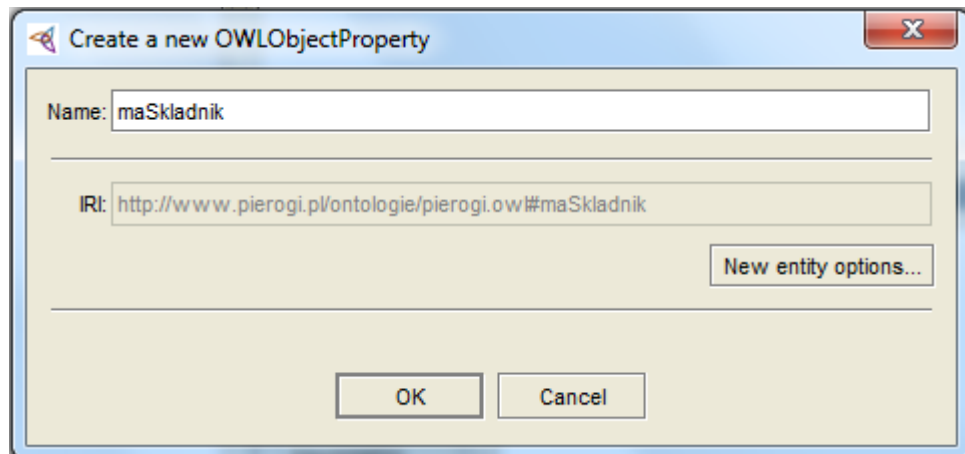
Rysunek 12.



Rysunek 13.

**Ćwiczenie 7. Tworzenie własności obiektowej *maSkładnik*.**

1. Wybierz zakładkę *Object Properties*. Kliknij główną własność *topObjectProperty*, a następnie użyj przycisku *Add sub property* do stworzenia nowej własności.
2. Nadaj nazwę własności *maSkładnik*, wykorzystując okienko dialogowe, takie jak na Rysunku 14.



Rysunek 14.



Mimo, iż nie ma konkretnej konwencji nazywania własności, to zaleca się stosować taką samą konwencję, jak w przypadku nazw klas. Ponadto zaleca się, aby własności zaczynały się od słów *has* albo *is*, w przypadku języka polskiego odpowiednio *ma* i *jest*.

Mając już dodaną własność *maSkladnik*, możemy dodać kolejne dwie własności: *maCiasto* i *maNadzienie*. W OWLu, własności mogą mieć podwłasności, więc możliwe jest tworzenie ich hierarchii. Podwłasności są bardziej szczegółowe od swoich nadrzędnych własności (w taki sam sposób, jak podklasy uszczegółwiają nadrzędne klasy). Na przykład własność *maMatke* może specjalizować własność *maRodzica*. W ten sposób dochodzimy do wniosku, że w naszej ontologii własności *maNadzienie* i *maCiasto* powinny być stworzone jako podwłasności *maSkladnik*. Jeżeli np. *maCiasto* będzie łączyć dwa różne obiekty, to oznacza, że te dwa obiekty są także połączone własnością *maSkladnik*.

#### **Ćwiczenie 8. Tworzenie *maCiasto* i *maNadzienie* jako podwłasności *maSkladnik*.**

---

1. Aby stworzyć własność *maNadzienie* jako podwłasność *maSkladnik*, zaznacz *maSkladnik* na drzewie hierarchii własności.
  2. Kliknij przycisk *Add sub property*.
  3. Nazwij nową własność jako *maNadzienie*.
  4. Powtórz powyższe kroki nazywając nową właściwość jako *maCiasto*.
- 

Zauważ, że jest także możliwe stworzenie podwłasności typu obiektowego. Jednak nie jest możliwe mieszanie własności obiektowych i własności literałowych, czyli np. próba stworzenie własności obiektowej jako podwłasności literałowej nie zostanie pomyślnie zakończona.

## 6. Dziedziny własności i ich zakresy (przeciwdziedziny).

Własności mogą mieć swoje dziedziny i zakresy. Własności łączą obiekty z dziedziny z obiektami z zakresu. Przykładowo w naszej ontologii własność *maNadzienie* będzie prawdopodobnie łączyć obiekty należące do klasy *Pierogi* z obiektami należącymi do klasy *PierogiNadzienie*. W tym przykładzie dziedzina własności *maNadzienie* to *Pierogi* a zakres to *PierogiNadzienie*.



Należy sobie uświadomić, że w języku OWL dziedziny i zakresy nie powinny być postrzegane jako ograniczenia do sprawdzenia. Jeśli własność *maNadzienie* ma ustawioną dziedzinę na *Pierogi* i zastosujemy tę własność do *Lody* (obiekty będące członkami klasy *Lody*), to nie powinno powodować żadnego błędu. Zamiast tego, zostanie to wykorzystane do wywnioskowania, że klasa *Lody* powinna być podklasą klasy *Pierogi*.

Chcemy teraz ustalić zakres własności *maNadzienie* jako *PierogiNadzienie*. Aby to zrobić należy wykorzystać widok przedstawiony na Rysunku 15.

### Ćwiczenie 9. Sprezycowanie zakresu własności *maNadzienie*.

---

1. Upewnij się, że własność *maNadzienie* jest zaznaczona w hierarchii własności.
  2. Kliknij *Add* (+) obok napisu *Ranges* w polu *Description* (Rysunek 15), następnie wybierz zakładkę *class hierarchy*. Pojawi się okienko umożliwiające wybór klasy z naszej hierarchii klas.
  3. Wybierz *PierogiNadzienie* i naciśnij *OK*. Klasa *PierogiNadzienie* powinna być teraz wyświetlana na liście zakresu własności.
- 



Rysunek 15.



NOTE

Możliwe jest ustalenie większej liczby klas jako zakresu dla własności. Jeśli więcej klas jest wyspecyfikowanych w Protégé 4, zakres własności jest interpretowany jako część wspólna klas.

### Ćwiczenie 10. Ustawienie klasy *Pierogi* jako dziedziny własności *maNadzienie*.

---

1. Upewnij się, że własność *maNadzenie* jest wybrana w hierarchii własności.
  2. Kliknij *Add* (+) obok napisu *Domains* w polu *Description*. Pojawi się okienko pozwalające na wybór odpowiedniej klasy.
  3. Wybierz *Pierogi* i kliknij *OK*. Klasa *Pierogi* powinna być teraz wyświetlana na liście dziedziny.
- 

#### MEANING



To oznacza, że obiekty znajdujące się „z lewej strony” własności *maNadzenie* będą wywiedzione jako obiekty klasy *Pierogi*. Dowolny obiekt będący „po prawej stronie” będzie wywiedziony jako obiekt klasy *PierogiNadzenie*. Na przykład, jeśli mamy obiekty *a* i *b* oraz asercję *a maNadzenie b*, to wniosek będzie taki, że *a* jest członkiem klasy *Pierogi* i *b* jest członkiem klasy *PierogiNadzenie*.

#### Ćwiczenie 11. Ustawienie dziedziny i zakresu dla własności *maCiasto*.

---

1. Wybierz własność *maCiasto*.
  2. Ustaw dziedzinę tej własności na *Pierogi*.
  3. Ustaw zakres własności na *PierogiCiasto*.
- 

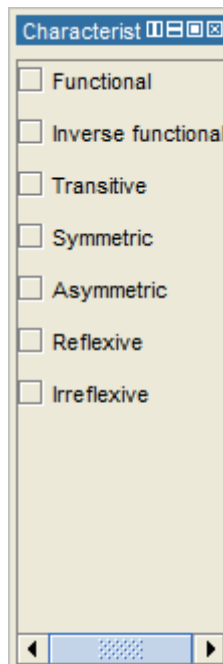


W tym tutorialu ustawialiśmy dziedziny i zakresy różnych własności, jednakże w rzeczywistości nie zawsze doradza się wykonywanie tych czynności. Fakt, że dziedziny i zakresy nie zachowują się jak ograniczenia i fakt, że mogą powodować niespodziewane wyniki klasyfikacji mogą prowadzić do problemów i nieoczekiwanych skutków ubocznych, a one mogą być trudne do znalezienia i poprawienia w dużej ontologii.

## 7. Wybrana charakterystyka obiektowych własności OWL.

OWL pozwala na wzbogacenie znaczenia własności poprzez użycie ich charakterystyki (Rysunek 16).



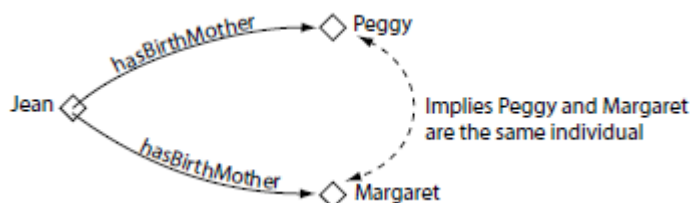


Rysunek 16.

Poniższe podrozdziały przedstawiają wybrane charakterystyki własności.

### 7.1 Własność funkcyjna

Jeśli własność jest własnością funkcyjną dla danego obiektu, to może istnieć co najwyżej jeden obiekt, z którym może być powiązany poprzez tę własność. Rysunek 17 przedstawia przykład własności funkcyjnej *hasBirthMother* – można mieć tylko jedną biologiczną matkę. Jeśli występuje asercja *Jean hasBirthMother Peggy*, i dodatkowo *Jean hasBirthMother Margaret*, to, ponieważ *hasBirthMother* jest własnością funkcyjną, możemy wywnioskować, że *Peggy* i *Margaret* musi być tym samym obiektem. W przeciwnym wypadku powyższe stwierdzenia byłyby ze sobą niezgodne.

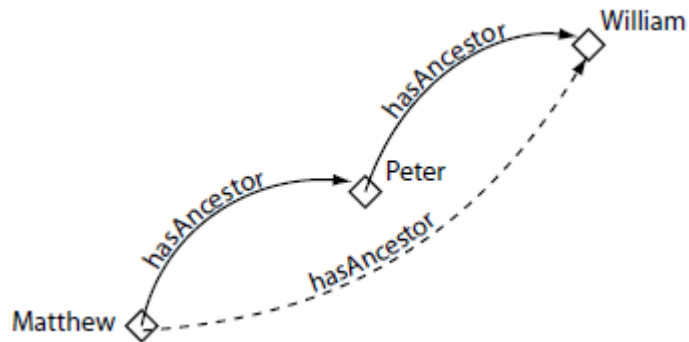


Rysunek 17.

### 7.2 Własności przechodnie.

Jeżeli własność jest przechodnia i łączy ona obiekt *a* z obiektem *b*, a także obiekt *b* z obiektem *c*, to można powiedzieć, że własność ta łączy obiekt *a* z obiektem *c*. Przykład pokazany jest na Rysunku 18. Jeżeli obiekt *Matthew* ma

przodka *Peter*, a on z kolei ma przodka *William*, to jest on także przodkiem *Matthew*.



Rysunek 18.

### Ćwiczenie 12. Oznaczenie *maSkładnik* jako własność przechodnią.

---

1. Wybierz własność *maSkładnik* z hierarchii własności.
  2. Na widoku *Characteristic* zaznacz *Transitive*.
- 



Jeżeli własność jest przechodnią, to nie może być funkcyjną.

Chcemy także, aby nasze pierogi miały tylko jeden rodzaj ciasta. Zrobimy to poprzez zmianę własności *maCiasto* na funkcyjną, dzięki czemu będzie mogła mieć tylko jedną wartość dla danego obiektu.

### Ćwiczenie 13. Oznaczenie *maCiasto* jako własność funkcyjną.

---

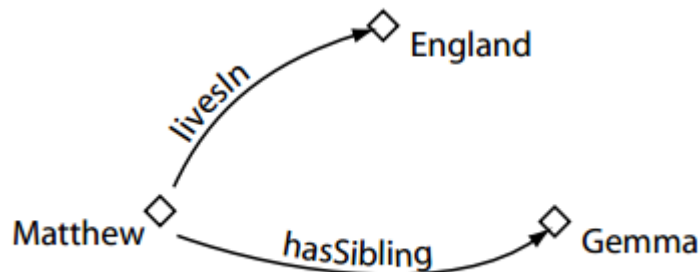
1. Wybierz własność *maCiasto* z hierarchii własności.
2. Na widoku *Characteristic* zaznacz *Functional*.

## 8. Opisywanie i definiowanie klas.

Mając stworzone własności możemy teraz użyć ich do definiowania i opisu klas naszej ontologii.

### 8.1 Ograniczenia własności

Jak pewnie wiesz w OWL własności opisują związki binarne. Własności literałowe (datatype) opisują związki pomiędzy obiektami a danymi. Własności obiektowe opisują relację pomiędzy dwoma obiektami. Rysunek 19 przedstawia dwie przykładowe własności; obiekt *Matthew* jest połączony z obiektem *Gemma* przez własność *hasSibling*. Możemy myśleć o takich obiektach jako o klasie obiektów które posiadają związek *hasSibling*. Główną ideą jest to, że klasy obiektów są opisywane/definiowane przez związki, w których są one stronami. W OWL możemy definiować takie klasy używając ograniczeń.



Rysunek 19.



*Ograniczenia* opisują klasy obiektów na podstawie związków, w jakich są obiekty będące częścią danej klasy. Innymi słowy *ograniczenie* jest rodzajem klasy, tak samo jak i klasy nazwane.

### Przykłady ograniczeń

Spójrzmy na kilka przykładów które pomogą nam wyjaśnić jakiego rodzaju klasy obiektów moglibyśmy opisywać na podstawie ich własności.

- Klasę obiektów które mają przynajmniej jeden związek *hasSibling*,
- Klasę obiektów które mają conajmniej jeden związek *hasSibling* z członkiem klasy *Man* - na przykład obiekty które mają chociaż jednego brata,
- Klasy obiektów które mają tylko związek *hasSibling* z obiektem który należy do klasy *Women* - są to obiekty które posiadają tylko siostry,
- Klasy obiektów które mają więcej niż trzy związki typu *hasSibling*,
- Klasy obiektów które mają przynajmniej jeden związek typu *maNadzenie* z obiektem, który należy do klasy *NadzenieMięsne*,
- klasy obiektów które mają związek *maNadzenie* z obiektami, które należą do klasy *NadzenieChińskie*.

W OWL możemy opisać wszystkie powyższe klasy obiektów z wykorzystaniem *ograniczeń*. Ograniczenia w OWL możemy podzielić na 3 główne kategorie:

- ograniczenia kwantyfikatorowe,
- ograniczenia ilościowe,
- ograniczenia "posiada wartość".

Na początku będziemy używać ograniczeń kwantyfikatorowych, które można podzielić na egzystencjalne i uniwersalne. Oba typy ograniczeń zilustrujemy w tym tutorialu.

#### Ograniczenia Egzystencjalne i Uniwersalne

- Ograniczenia egzystencjalne opisują klasy indywiduów które są stroną w *conajmniej jednym* związku określoną własnością z obiektem, który jest instancją określonej klasy. Na przykład: klasa indywiduów które mają *conajmniej jeden* (some) związek *maNadzenie* z instancją klasy *NadzenieChińskie*. W Protégé 4 słowo kluczowe "**some**"(jakiś) jest używane do wskazania ograniczenia egzystencjalnego.
- Ograniczenie uniwersalne opisuje klasy indywiduów które dla określonej własności mają związki tylko z obiektami należącymi do określonej klasy. Na przykład: klasa obiektów które mają związek *maNadzenie* tylko z obiektami klasy *NadzenieChińskie*.

Przyjrzyjmy się bliżej przykładowi ograniczeń egzystencjalnych. Ograniczenie *maNadzenie some* (jakiś) *NadzenieChińskie* jest ograniczeniem egzystencjalnym (wskazuje na to zastosowanie słowa kluczowego "some"/"jakiś"), które jest wprowadzone za pomocą własności *maNadzenie*, która ma "wypełnienie" (wartość zakresu) *NadzenieChińskie*. To ograniczenie opisuje *klasę*, indywiduów które mają *conajmniej jeden* związek *maNadzenie* z obiektem, który jest członkiem klasy *NadzenieChińskie*.

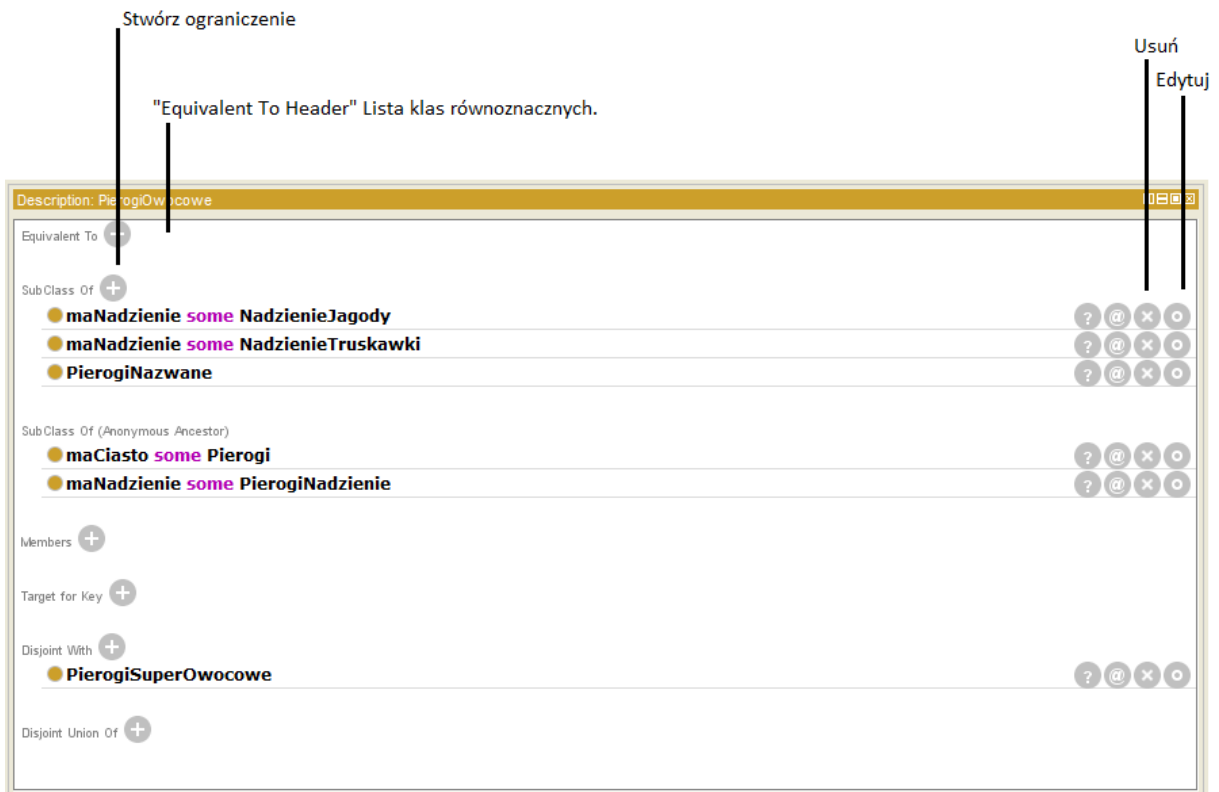
#### MEANING



Ograniczenie opisuje *klasę anonimową* (nienazwaną klasę). Klasa anonimowa zawiera wszystkie obiekty, które spełniają dane ograniczenie - na przykład: wszystkie obiekty, które mają dany związek należą do danej klasy.

Ograniczenia klas są wyświetlane i edytowane za pomocą widoku "*Class Description View*", który jest przedstawiony na rysunku 20. Widok "*Class Description View*" jest "sercem" zakładki "*Classes*" w Protégé i wirtualnie przechowuje wszystkie informacje użyte do opisu klasy. Na pierwszy rzut oka, ekran "*Class Description View*" może wydawać się skomplikowany, jednakże stanie się jasne, że jest to niezwykle skuteczny sposób opisywania i definiowania klas.

Ograniczenia są używane w opisie klas OWL do tworzenia anonimowych nadklas opisywanych klas.



Rysunek 20.

## 8.2 Ograniczenia egzystencjalne

Ograniczenia egzystencjalne są zdecydowanie najczęstszym typem ograniczeń wykorzystywanym w ontologiach OWL. Ograniczenie egzystencjalne opisuje klasę indywiduów które mają *co najmniej jeden* ("some"/"jakiś") związek poprzez określoną własność z indywiduum, który jest członkiem określonej klasy. Na przykład, *maCiasto some PierogiCiasto* opisuje wszystkie obiekty, które mają *co najmniej jeden* związek poprzez własność *maCiasto* z obiektem, który jest członkiem klasy *PierogiCiasto* - mówiąc prościej - wszystkie obiekty, które mają *co najmniej jedno* ciasto.

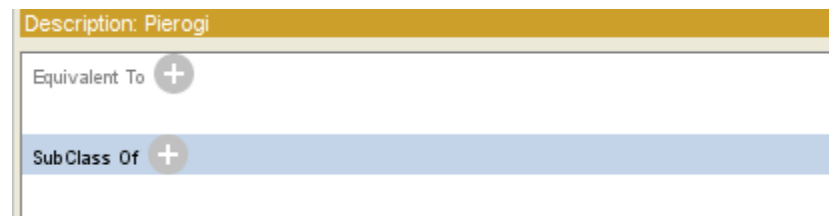


*Ograniczenia egzystencjalne* są znane również jako ograniczenia *co najmniej jeden z*.

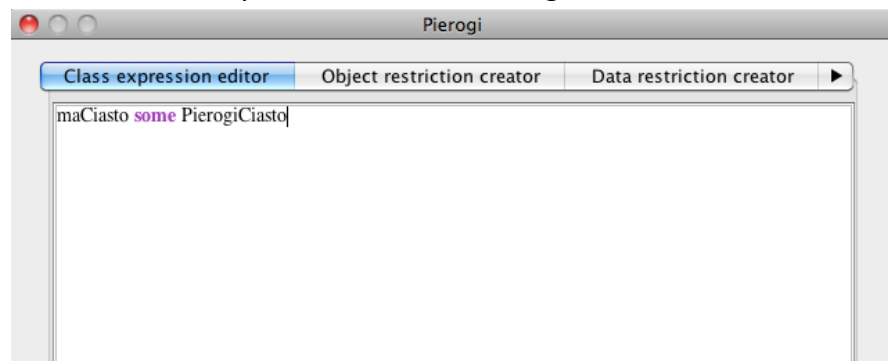
#### Ćwiczenie 14. Dodanie ograniczenia do klasy *Pierogi*, stanowiącego o tym, że *Pierogi* muszą mieć *PierogiCiasto*

---

1. Wybierz *Pierogi* z "class hierarchy" na zakładce "Classes".
  2. Wybierz przycisk "Add" (+) znajdujący się obok nagłówka "Sub Class of" w "Class Description View", przedstawionym na rysunku 21, w celu stworzenia potrzebnego warunku.
  3. Wybierz zakładkę "Class expression editor" - spowoduje to wyświetlenie pola tekstowego gdzie wpisujemy nasze ograniczenie, tak jak przedstawiono to na rysunku 22.
- 



Rysunek 21. Tworzenie ograniczenia



Rysunek 22. Tworzenie ograniczenia

Pole tekstowe w zakładce "Class expression editor" pozwala Ci na tworzenie ograniczeń używając nazw klas, własności i indywiduów.

Aby stworzyć ograniczenie musimy wykonać trzy kroki:

- wprowadzić nazwę własności, która ma być ograniczona (lub wybrać ją z listy "property list"),
- wprowadzić typ ograniczenia (np. "some" gdy chcemy stworzyć ograniczenie egzystencjalne),
- wprowadzić "filler"(klasę / indywiduum, która / które ma być w związku z klasą, dla której definiujemy to ograniczenie).

## Ćwiczenie 15. Dodanie ograniczenia do klasy *Pierogi*, stanowiącego o tym, że *Pierogi* muszą mieć *PierogiCiasto* (kontynuacja)

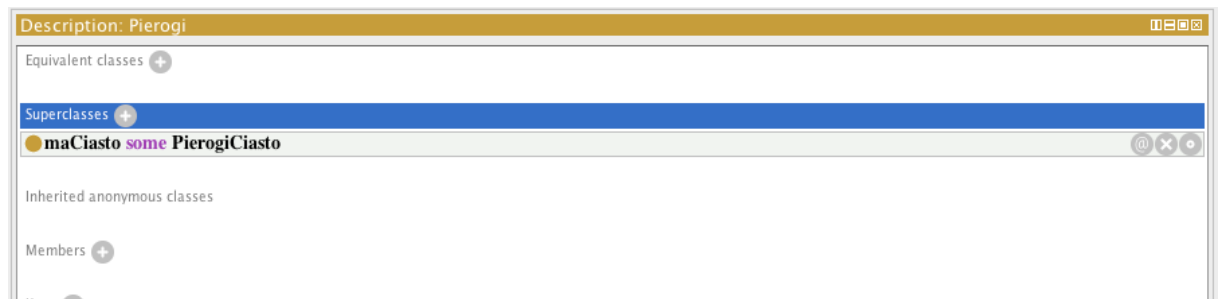
---

1. Możesz korzystając z "drag and drop" przeciągnąć *maCiasto* z listy "property list" na pole tekstowe edytora lub po prostu je wpisać,
  2. Teraz dodaj typ ograniczenia, w tym przypadku jest to ograniczenie egzystencjalne - należy więc wpisać słowo kluczowe "some",
  3. Uzupełnij "filler" jako *PierogiCiasto* - aby to zrobić wprowadź *PierogiCiasto* lub skorzystaj z opcji "drag and drop".
  4. Wciśnij przycisk *OK* aby utworzyć ograniczenie i zamknąć edytor. Jeżeli wszystkie informacje zostały poprawnie wprowadzone edytor zamknie się a wprowadzone przez nas ograniczenie zostanie przedstawione pod nagłówkiem "*SubClass of*". Jeśli wpisane zostały błędne dane będą one podkreślone czerwonym kolorem i okienko z błędem, które się pojawi będzie zawierało podpowiedź co jest źródłem błędu.
- 



Bardzo użyteczną funkcją "expression builder'a" jest możliwość automatycznego uzupełniania nazw klas, własności oraz obiektów. Automatyczne uzupełnianie jest aktywowane poprzez wciśnięcie "Ctrl-Space".

Widok "class description" powinien wyglądać teraz podobnie do tego przedstawionego na rysunku 23.



Rysunek 23. Widok "class description": Opis klasy *Pierogi*.



Opisaliśmy klasę *Pierogi* jako podklasę *Thing* oraz podklasę obiektów, które mają ciasto, które jest rodzajem *PierogiCiasto*. Zauważmy, że jest to warunek konieczny – jeżeli coś jest *Pierogiem* to jest konieczne, żeby było członkiem klasy *Thing* (w OWL wszystko jest członkiem klasy *Thing*) i jest konieczne, aby posiadało jakiś rodzaj *PierogiCiasto*.

Bardziej formalnie, jeżeli coś jest obiektem klasy *Pierogi* to jest konieczne, aby był on w relacji z członkiem klasy *PierogiCiasto* poprzez własność *maCiasto*.

#### MEANING



Kiedy do opisu klas używamy ograniczeń, to właściwie określamy anonimowe nadklasy klas opisywanych. Na przykład, możemy powiedzieć, że *PierogiChińskie* są podklasą, między innymi, klasy *Pierogi* i również podklasą rzeczy, które mają co najmniej jedno ciasto będące *NadzieniemChińskim*.

#### Tworzenie różnych rodzajów Pierogów.

Nadszedł czas, aby dodać różne rodzaje pierogów do naszej ontologii. Zaczniemy od dodania *PierogiOwocowe*, które są pierogami z nadzieniem z truskawek i jagód. Aby utrzymać porządek naszej ontologii, pogrupujemy nasze różne pierogi w klasie *NazwanePierogi*.

#### Ćwiczenie 16. Tworzenie podklasy klasy *Pierogi* - *NazwanePierogi* i podklasy klasy *NazwanePierogi* - *PierogiOwocowe*

---

1. Wybierz zakładkę „Classes” i z hierarchii klas wybierz klasę *Pierogi*
  2. Wybierz przycisk "Add" ( + ), aby stworzyć nową podklasę klasy *Pierogi*, i nazwij ją *PierogiNazwane*.
  3. Stwórz nową podklasę klasy *PierogiNazwane* i nazwij ją *PierogiOwocowe*
  4. Dodaj komentarz do klasy *PierogiOwocowe* używając widoku „Annotations”, który jest zlokalizowany obok widoku hierarchii klas. Możesz wpisać: *Pierogi*, które mają nadzienie z truskawek i jagód. Pamiętajmy, że zawsze dobrze jest dokumentować klasy, własności itp., zwłaszcza, jeżeli będziemy później wykorzystywać je do budowy innych, bardziej, rozbudowanych ontologii.
- 

Mając stworzoną klasę *PierogiOwocowe* musimy teraz zdefiniować nadzienie dla tych pierogów. W tym celu dodamy dwa ograniczenia mówiące, że *PierogiOwocowe* mają nadzienie z *NadzienieTruskawki* i *NadzienieJagody*.

#### Ćwiczenie 17. Utworzenie ograniczenia egzystencjalnego na *PierogiOwocowe*, które będzie posiadało własność *maNadzienie*, z ograniczeniem na *NadzienieTruskawki*, stanowiącym, że *PierogiOwocowe* mają co najmniej jedno nadzienie *TruskawkiNadzienie*.

---

1. Upewnij się, że zaznaczona jest klasa *PierogiOwocowe* w hierarchii klas.
2. Wybierz przycisk "Add" ( + ) sekcji *Subclass Of* w widoku „Class Description view”.



3. Wybierz zakładkę „Object restriction creator”.
  4. Wybierz *maNadzenie* w polu „Restricted property”
  5. Wybierz *some* jako „Restriction type”
  6. Wybierz klasę *NadzenieTruskawki* jako “Restriction filler”
  7. Kliknij *OK* aby stworzyć ograniczenie – jeżeli pojawią się jakiegokolwiek błędy, ograniczenie nie zostanie stworzone.
- 

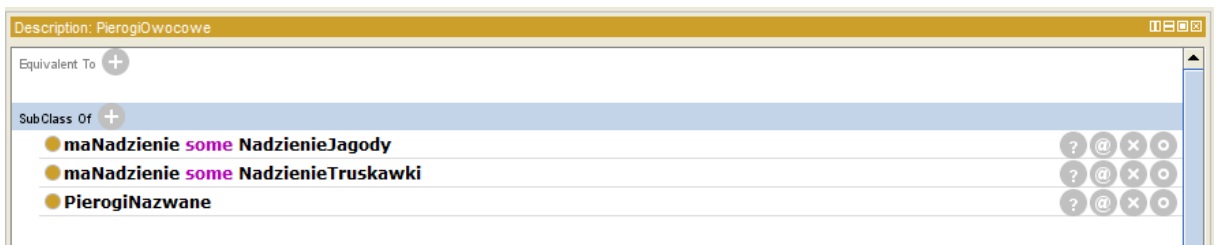
Teraz dodamy jeszcze *NadzenieJagody* do *PierogiOwocowe*.

**Ćwiczenie 18. Stworzenie ograniczenia egzystencjalnego na *PierogiOwocowe*, które będzie posiadało własność *maNadzenie*, z ograniczeniem na *NadzenieJagody*, stanowiącym, że *PierogiOwocowe* mają co najmniej jedno nadzenie *JagodyNadzenie*.**

---

1. Upewnij się, że zaznaczona jest klasa *PierogiOwocowe* w hierarchii klas.
  2. Wybierz przycisk "Add" ( + ) sekcji *Subclass Of* w widoku „Class Description view”.
  3. Wybierz zakładkę „Object restriction creator”.
  4. Wybierz *maNadzenie* w polu „Restricted property”
  5. Wybierz *some* jako „Restriction type”
  6. Wybierz klasę *NadzenieJagody* jako “Restriction filler”
  7. Kliknij *OK* aby stworzyć ograniczenie – jeżeli pojawią się jakiegokolwiek błędy, ograniczenie nie zostanie stworzone.
- 

Sekcja „Description” powinna wyglądać jak na rysunku 24.



Rysunek 24. Opis *PierogiOwocowe*

Teraz stworzymy klasę reprezentującą *Pierogi Super Owocowe*, które posiadają nadzenie z truskawek, jagód i awokado. Ponieważ klasa *PierogiSuperOwocowe* jest bardzo podobna do klasy *PierogiOwocowe* (*Pierogi Super Owocowe* są prawie takie

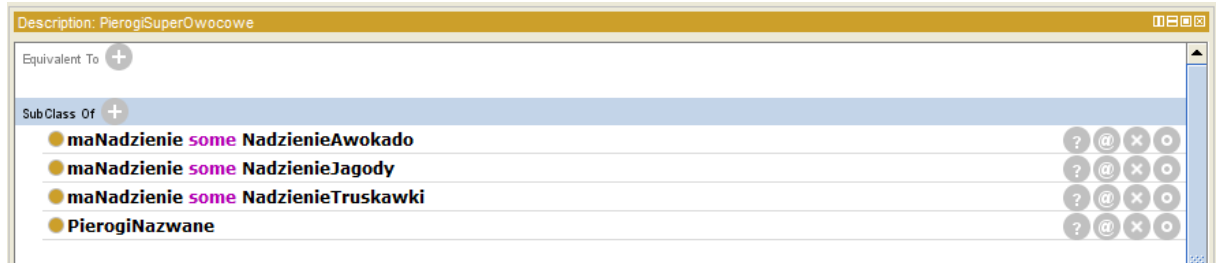
same, jedyną różnicą jest to, że Pierogi Super Owocowe mają jeden dodatkowy składnik nadzienia - awokado) stworzymy kopię klasy *PierogiOwocowe* i dodamy dodatkowe ograniczenie, które opisze, że mają jeszcze dodatkowe nadzienie z awokado.

### Ćwiczenie 19. Stworzenie klasy *PierogiSuperOwocowe* jako kopii i modyfikacji klasy *PierogiOwocowe*

---

1. Wybierz zakładkę „Classes” i z hierarchii klas wybierz klasę *PierogiOwocowe*
  2. Wybierz „Duplicate selected class” z menu *Edit*. W oknie, które się pojawi, należy wpisać nazwę nowej klasy. Wpisujemy *PierogiSuperOwocowe*.
  3. Upewnij się, że klasa *PierogiSuperOwocowe* jest wybrana. Teraz wybierz przycisk "Add" ( + ) sekcji *Subclass Of* w widoku „Class Description view”
  4. Przejdź na zakładkę „Object restriction creator”.
  5. Wybierz *maNadzienie* w polu „Restricted property”
  6. Wybierz *some* jako „Restriction type”
  7. Wybierz klasę *NadzienieAwokado* jako „Restriction filer”
  8. Kliknij *OK* aby stworzyć ograniczenie.
- 

Sekcja „Description” powinna wyglądać jak na rysunku 25.



Rysunek 25. Opis *PierogiSuperOwocowe*

### Ćwiczenie 20. Stworzenie klasy *PierogiSerowoOwocowe*

---

*PierogiSerowoOwocowe* są również praktycznie takie same jak *PierogiOwocowe*, ale mają dodatkowe nadzienie z sera. Utwórz *PierogiSerowoOwocowe* jako kopia klasy *PierogiOwocowe* i dodaj odpowiednie ograniczenie (tzn. dodatkowe nadzienie *NadzienieMozzarella*)

---

Sekcja ‘SubClass Of’ w ‘Description’ pozwala na podanie warunku koniecznego jakie muszą spełniać instancje danej klasy. Jest to częściowa definicja klasy. Możliwe

jest także zamodelowanie pełnej *definicji* klasy za pomocą sekcji *'Equivalent To'*, która umożliwi podanie warunku *koniecznego i wystarczającego* do tego aby spełniający go obiekt znalazł się w danej klasie.

### Ćwiczenie 21. Utworzenie definicji klasy *PierogiSerowe*

---

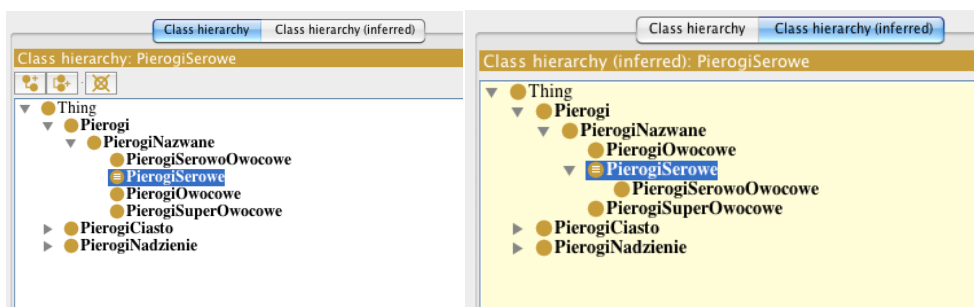
1. Stwórz nową podklasę klasy *PierogiNazwane* i nazwij ją *PierogiSerowe*
  2. Upewnij się, że klasa *PierogiSerowe* jest wybrana. Teraz wybierz przycisk "Add" (+) sekcji *Equivalent To* w widoku „*Class Description view*”
  3. Wybierz zakładkę „*Object restriction creator*”.
  4. Wybierz *maNadzenie* w polu „*Restricted property*”
  5. Wybierz *some* jako „*Restriction type*”
  6. Wybierz klasę *NadzenieSer* jako „*Restriction filler*”
  7. Kliknij *OK* aby stworzyć ograniczenie.
- 

## 10. Wnioskowanie

### Ćwiczenie 22. Automatyczna klasyfikacja/tworzenie hierarchii klas na podstawie ich własności.

---

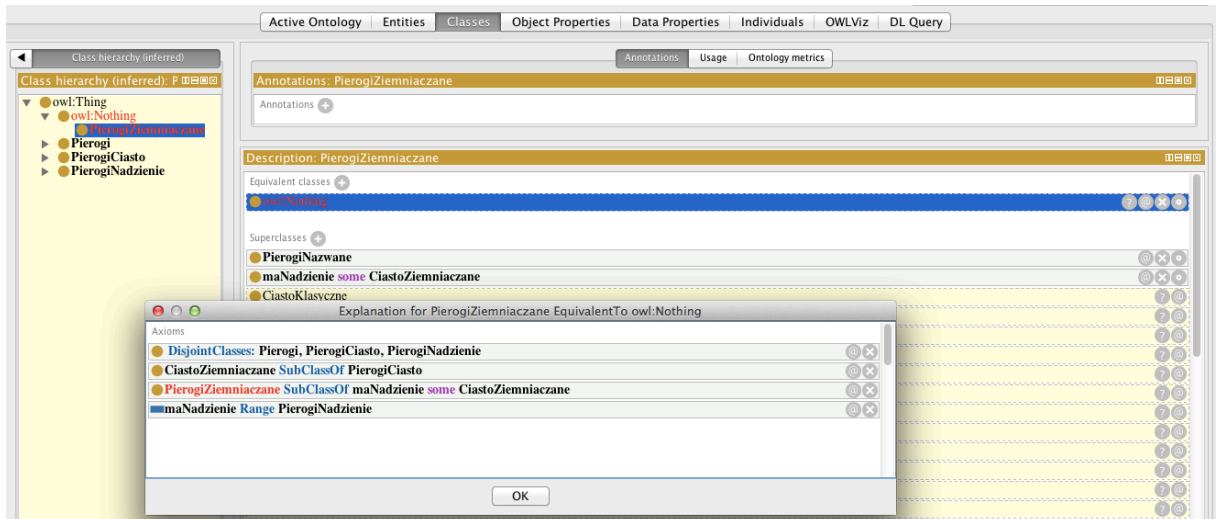
1. Z menu *'Reasoner'* wybierz *'Start reasoner'* (zostaw zaznaczony domyślny silnik wnioskujący lub wybierz dowolny jeżeli żaden nie jest zaznaczony). Silnik wnioskujący dokona automatycznej (re-)klasyfikacji hierarchii klas.
  2. Porównaj hierarchię klas jaką wprowadziłeś (*'Class hierarchy'*) z tą jaką wywnioskował silnik (*'Class hierarchy (inferred)'*). Zauważ, że w tym drugim przypadku, klasa *PierogiSerowoOwocowe* została umieszczona jako podklasa klasy *PierogiSerowe* (rys. 26).
- 



Rysunek 26. Porównanie wprowadzonej i automatycznie wywnioskowanej hierarchii klas.

## Ćwiczenie 23. Automatyczna klasyfikacja – wykrywanie niespójności.

1. Wprowadź nową podklasę klasy *PierogiNazwane* o nazwie *PierogiZiemniaczane*.
2. Wprowadź dla tej klasy ograniczenie własności o postaci *maNadzenie some CiastoZiemniaczane*.
3. Ponownie uruchom silnik wnioskujący (z menu 'Reasoner' wybierz 'Start reasoner' bądź też 'Synchronize reasoner').
4. Silnik wnioskujący powinien wykryć niespójność z uwagi na rozłączność klas *Nadzenie* i *Ciasto*. W wyniku tej niespójności, klasa *PierogiZiemniaczane* staje się niespełnialna (nie ma takich instancji, które mogłyby spełnić ograniczenia nałożone na tą klasę, są one sprzeczne). Klasa *PierogiZiemniaczane* zostaje oznaczona na czerwono i w 'Class hierarchy' (inferred) staje się podklasą klasy *Nothing*, oznaczającej niespełnialne klasy.
5. W widoku 'Description' klasy *PierogiZiemniaczane*, klasa *Nothing* pojawia się jako 'Equivalent Class'.
6. Po kliknięciu ikonki ze znakiem zapytania, można odczytać wyjaśnienie takiej dedukcji (patrz Rysunek 27).



Rysunek 27. Ilustracja niespełnialnej klasy *PierogiZiemniaczane* i objaśnienia takiej dedukcji o niespełnialności.