

PYTHON

DOKUMENTACJA

Grzegorz Janosz & Michał Szczepaniak

Spis treści

KLASA DICTLIB	4
SPOSÓB INICJALIZACJI	4
METODY	4
<i>save()</i>	4
<i>load()</i>	4
<i>delete()</i>	4
<i>find()</i>	5
<i>get_lexeme()</i>	5
KLASA MULTISEGMENT	5
KLASA LEXEME (LEKSEM)	6
ATRYBUTY	6
<i>self.basic_form</i>	6
<i>self.flectional_label</i>	6
<i>self.label</i>	6
<i>self.flection</i>	6
<i>self.multi_segments</i>	6
METODY	6
<i>find_flection_enums()</i>	6
FIND_FLECTION()	6
KLASA NOUNLEXEME	7
ATRYBUTY	7
<i>self.is_gerundive</i>	7
<i>self.verb_data</i>	7
METODY	7
<i>get_verb_data()</i>	7
KLASA VERBLEXEME	7
ATRYBUTY	7
<i>self.participles</i>	7
<i>self.has_gerundive</i>	7
<i>self.gerundive_data</i>	7
METODY	7
<i>get_gerundive_data()</i>	7
KLASA ADJECTIVELEXEME	8
ATRYBUTY	8
<i>self.is_gradable</i>	8
<i>self.is_participle</i>	8
<i>self.my_grade</i>	8
<i>self.grades</i>	8
<i>self.verb_data</i>	8
<i>self.participle_kind</i>	8
METODY	8
<i>get_grades()</i>	8
<i>get_verb_data()</i>	8
<i>get_participle_kind()</i>	8

KLASA ADVERBLEXEME	9
Atrybuty	9
<i>self.is_gradable</i>	9
<i>self.grades</i>	9
KLASA UNINFLECTEDLEXEME	9
Atrybuty	9
<i>self.is_participle</i>	9
<i>self.verb_data</i>	9
<i>self.participle_kind</i>	9
Metody	9
<i>get_verb_data()</i>	9
<i>get_participle_kind()</i>	9
POZOSTAŁE PODKLASY	10
KLASA FILTERSTRUCTURE	10
KLASA LABELS	10
Enumy	10
Metody:	10
<i>get_label_from_flectional_label</i>	10
KLASA NOUN(ENUM)	11
KLASA VERB(ENUM)	12
KLASA ADJECTIVE(ENUM)	13
KLASA NUMERAL(ENUM)	14
KLASA PRONOUN(ENUM)	15
KLASA ADVERB(ENUM)	15
POZOSTAŁE	15

Klasa DictLib

Sposób inicjalizacji

Kod:

```
files = ["../files/pospolite (1).txt", "../files/adj.txt", "../files/WS tylko rzecz.txt", "../files/adv.txt",  
        "../files/im_nom.txt"]  
file_types = [0, 1, 2, 1, 1]  
bt = DictLib(files, file_types)
```

czyli DictLib(files, file_types) – gdzie:

files – lista plików do sparsowania

file_types – lista zawierająca typ pliku dla każdego pliku w **files**. Typy plików:

- 0 – plik ze zwykłą regularną odmianą
- 1 – filtr morfologiczny
- 2 – plik z wielosegmentowcami

Utworzenie DictLib z 9 milionami słów wraz z zapisaniem do pliku (save) na komputerze z 16 GB ramu i 8-rdzeniowym procesorem Intel® Core(TM) i7 2.8GHz zajmuje troszkę mniej niż 2 minuty.

Metody

save()

Nagłówek:

```
def save(self, file_name='DictLib')
```

Zapisuje DictLib do pliku o podanej nazwie. Po zapisaniu pliku, można go ponownie załadować metodą load(). Metoda sama dopisuje rozszerzenie .pickle

load()

Nagłówek:

```
def load(file_name='DictLib'):
```

Ładuje zapisany DictLib z pliku. Nazwa podana tutaj ma być ta sama co przy save() – domyślnie DictLib.

delete()

Nagłówek:

```
def delete(file_name='DictLib'):
```

Usuwa plik utworzony przez save()

find()

Nagłówek:

```
def find(self, word):
```

Najważniejsza metoda DictLib. Wyszukuje podane słowo w drzewie i zwraca listę leksemów. Jeśli nie znajdzie żadnego słowa zwraca pustą listę oraz drukuje odpowiedni komunikat:

```
zapis
Word "zapis" not found!
```

Gdy znajdzie jakieś słowa zwraca je odpowiadające im leksemy w postaci obiektów Lexeme (opisanych poniżej). Np. dla `find(„biel”)` zwraca leksem od **czasownika** *bielić* (jest to forma 2 os. l. poj. trybu rozkazującego tego czasownika) oraz leksem **rzeczownika** *biel*.

Jeśli znajdzie słowa tylko w wielosegmentowach zwraca obiekt Lexeme z następującymi parametrami – czyli tylko atrybut **multi_segments** posiada odpowiednią zawartość.

```
else:
    self.basic_form = "None"
    self.flectional_label = "None"
    self.label = "None"
    self.flection = []
    self.multi_segments = [MultiSegment(multi_segment) for multi_segment in multi_segments]
```

get_lexeme()

Nagłówek:

```
def get_lexeme(self, lexeme_data):
```

Parametr **lexeme_data** to tupla w następującej postaci: **(word, flectional_label)** gdzie **word** to słowo, a **flectional_label** to jego etykieta fleksyjna. Metoda ta zwraca jeden konkretny leksem – para słowo – etykieta fleksyjna jest unikatowa. Takie tuple są zwracane w metodach w obiektach **Lexeme** stąd właśnie taka implementacja.

Klasa MultiSegment

Posiada atrybut **self.line** przechowujący linijkę z pliku z wielosegmentowacami oraz **__repr__** zamieniający ją w czytelną formę.

Klasa Lexeme (Leksem)

Bazowa klasa – posiada subklasy dla każdego rodzaju etykiety fleksyjnej. W bazowej (a więc i w subklasach) są zawarte następujące atrybuty i metody:

Atrybuty

self.basic_form

String z bazową formą czasownika (pierwsza kolumna z pliku z odmianą)

self.flectional_label

Etykieta fleksyjna

self.label

Etykieta z enuma Label (patrz niżej), czyli czy rzeczownik, czasownik itd.

self.flection

Słownik zawierający wszystkie dostępne odmiany danego słowa – to jakie słowa zawiera różni się dla konkretnych rodzajów podklas. Ogólnie jeśli klasa zawiera atrybut będący słowem, to to słowo także zawiera się we **flection** – np. przymiotnik stopniowalny będzie we flection zawierał pozostałe stopnie, a czasownik wszystkie swoje imiesłowy i ewentualne gerundium.

Kluczami są enumy (**Noun**, **Verb** itd.) a wartościami stringi z odpowiedziami słowami.

self.multi_segments

Lista obiektów typu **MultiSegment**

Metody

find_flection_enums()

Nagłówek:

```
def find_flection_enums(self, searched_word):
```

dla danego słowa **searched_word** zwraca listę enumów w jakich odmianach występuje.

Np. dla **get_lexeme**("zjedzenie", "ABCA").**find_flection_enums**("zjedzenie") zwraca listę [Noun.Singular_Nominative, Noun.Singular_Accusative, Noun.Singular_Vocative].

find_flection()

Nagłówek:

```
def find_flection(self, searched_enum):
```

Dla podanego enuma formy (jednego z np. **Noun**, **Verb**) zwraca konkretną formę.

Przykład: dla leksemu rzeczownika "zły" **find_flection**(Noun.Singular_Accusative) (biernik liczby pojedynczej) zwróci „złego”.

Jeśli spróbujemy wyszukać np. w przymiotniku formę czasownika metoda wyrzuci błąd **ValueError**.

Dopuszczalne jest jedyne szukanie **Verb.Infinitive** (bezokolicznik czasownika) w rzeczownikach (mogą być odczasownikowe), przymiotnikach (mogą być imiesłowami przymiotnikowymi) oraz nieodmiennych (etykieta G – tutaj łądują imiesłowy przysłówkowe).

Jeśli enum jest dopuszczalny, ale dana forma nie istnieje to zwracany jest **None**.

Dla wyszukiwania stopni wyższych i najwyższych dla przymiotnika i przysłówka lepiej używać odpowiednich metody w odpowiadających im leksemach.

Klasa NounLexeme

Rzeczownik

Atrybuty

self.is_gerundive

Jest to bool o wartości **True** jeśli rzeczownik jest gerundium (rzeczownik odczasownikowy).

self.verb_data

Atrybut istnieje tylko dla rzeczowników będących gerundium. Istnieje getter.

Metody

get_verb_data()

Bezparametrowy – sprawdza czy rzeczownik jest gerundium – jeśli tak to zwraca tuple (**word**, **flectional_label**) czasownika od którego pochodzi (tupla potrzebna jest by uzyskać potrzebny leksem przy pomocy DictLib.get_lexeme(lexeme_data)).

Klasa VerbLexeme

Czasownik

Atrybuty

self.participles

Słownik z imiesłowami (para enum i string – np. **Verb.Present_Adverbial_Participant** : „jedząc”). Gdy nie ma żadnego imiesłowu, słownik jest pusty. By uzyskać dany imiesłów w zasadzie wystarczy wywołać metodę get na słowniku z imiesłowami (czyli właśnie **self.participles**)

self.has_gerundive

Jest to bool o wartości **True**, gdy czasownik posiada gerundium.

self.gerundive_data

Atrybut istnieje tylko gdy czasownik posiada gerundium. Istnieje osobny getter.

Metody

get_gerundive_data()

Bezparametrowy – sprawdza czy czasownik ma gerundium – jeśli tak to zwraca tuple (**word**, **flectional_label**) tego rzeczownika, tupla potrzebna jest by uzyskać potrzebny leksem przy pomocy DictLib.get_lexeme(lexeme_data).

Klasa AdjectiveLexeme

Przymiotnik

Atrybuty

self.is_gradable

Jest to bool o wartości **True**, gdy przymiotnik jest stopniowalny.

self.is_participle

Jest to bool o wartości **True**, gdy przymiotnik jest imiesłowem.

self.my_grade

Domyślnie **Adjective.Positive_Form** (stopień równy). Jeśli jednak jest którymś z wyższych stopni, wartość ta zmienia się.

self.grades

Słownik z pozostałymi stopniami (para enum i string – np. **Adjective.Positive_Form** : „wysoki”). Np. dla stopnia wyższego będzie posiadać stopień równy i najwyższy. Gdy przymiotnik nie jest stopniowalny, atrybut nie istnieje – dlatego istnieje dla niego getter.

self.verb_data

Atrybut istnieje tylko gdy przymiotnik jest imiesłowem – jest to tupla (**word**, **flectional_label**) czasownika tego imiesłowu. Istnieje getter.

self.participle_kind

Atrybut istnieje tylko gdy przymiotnik jest imiesłowem – zawiera enum oznaczający jaki to imiesłów, czyli jeden z dwóch: **Verb.Active_Adjectival_Participe**, **Verb.Passive_Adjectival_Participe**. Także ma getter.

Metody

get_grades()

Bezparametrowy – sprawdza czy przymiotnik jest stopniowalny – jeśli tak to zwraca słownik z tuplami pozostałych stopni przymiotnika.

get_verb_data()

Bezparametrowy – sprawdza czy przymiotnik jest imiesłowem – jeśli tak to zwraca tuplę (**word**, **flectional_label**) czasownika od którego pochodzi (tupla potrzebna jest by uzyskać potrzebny leksem przy pomocy **DictLib.get_lexeme(lexeme_data)**).

W przeciwnym przypadku zwraca **None**.

get_participle_kind()

Bezparametrowy – sprawdza czy przymiotnik jest imiesłowem – jeśli tak to zwraca enum oznaczający jaki to imiesłów, czyli jeden z dwóch: **Verb.Active_Adjectival_Participe**, **Verb.Passive_Adjectival_Participe**.

W przeciwnym przypadku zwraca **None**.

Klasa AdverbLexeme

Przysłówek.

Atrybuty

self.is_gradable

Jest to bool o wartości **True**, gdy przysłówek jest stopniowalny.

self.grades

Słownik z wszystkimi stopniami przysłówka (para enum i string – np. **Adverb.Positive_Form** : „wysoko”). Pozostałe stopnie przysłówka nie istnieją jako osobne rekordy, zatem **self.grades** istnieje dla każdego przysłówka i dla każdego zawiera stopień równy. Gdy przysłówek jest stopniowalny zawiera także pozostałe stopnie.

Klasa UninflectedLexeme

Nieodmienne – etykieta G

Atrybuty

self.is_participle

Jest to bool o wartości **True**, gdy leksem jest imiesłowem.

self.verb_data

Atrybut istnieje tylko gdy leksem jest imiesłowem – jest to tupla (**word**, **flectional_label**) czasownika tego imiesłowu. Istnieje getter.

self.participle_kind

Atrybut istnieje tylko gdy leksem jest imiesłowem – zawiera enum oznaczający jaki to imiesłów, czyli jeden z dwóch: **Verb.Present_Adverbial_Participale**, **Verb.Perfect_Adverbial_Participale**. Także ma getter.

Metody

get_verb_data()

Bezparametrowy – sprawdza czy leksem jest imiesłowem – jeśli tak to zwraca tuplę (**word**, **flectional_label**) czasownika od którego pochodzi (tupla potrzebna jest by uzyskać potrzebny leksem przy pomocy **DictLib.get_lexeme(lexeme_data)**).

W przeciwnym przypadku zwraca **None**.

get_participle_kind()

Bezparametrowy – sprawdza czy leksem jest imiesłowem – jeśli tak to zwraca enum oznaczający jaki to imiesłów, czyli jeden z dwóch: **Verb.Present_Adverbial_Participale**, **Verb.Perfect_Adverbial_Participale**.

W przeciwnym przypadku zwraca **None**.

Pozostałe podklasy

NumeralLexeme, PronounLexeme, TextLexeme, AcronymLexeme - brak dodatkowych atrybutów i metod

Klasa FilterStructure

Pomocnicza klasa, tworząca struktury dla różnych rodzajów filtrów pomagające w tworzeniu leksemów.

Klasa Labels

Jest to enum podstawowych rodzajów słów – po jednym dla każdej etykiety.

Enumy

```
NOUN = "A"  
VERB = "B"  
ADJECTIVE = "C"  
NUMERAL = "D"  
PRONOUN = "E"  
ADVERB = "F"  
UNINFLECTED = "G"  
TEXT = "H"  
ACRONYM = "I"
```

Metody:

get_label_from_flectional_label

```
@staticmethod  
def get_label_from_flectional_label(flectional_label):  
    label_string = flectional_label.strip()  
    label_string = flectional_label.strip('*')[0]  
    return Labels(label_string)
```

dla podanej etykiety fleksyjnej np “*CAAB”, zwraca odpowiedni enum (w tym przypadku ADJECTIVE).

Klasa Noun(Enum)

Enumy dla rzeczownika

```
Singular_Nominative = "Liczba pojedyncza, Mianownik"  
Singular_Genitive = "Liczba pojedyncza, Dopełniacz"  
Singular_Dative = "Liczba pojedyncza, Celownik"  
Singular_Accusative = "Liczba pojedyncza, Biernik"  
Singular_Instrumental = "Liczba pojedyncza, Narzędnik"  
Singular_Locative = "Liczba pojedyncza, Miejscownik"  
Singular_Vocative = "Liczba pojedyncza, Wołacz"  
Plural_Nominative = "Liczba mnoga, Mianownik"  
Plural_Genitive = "Liczba mnoga, Dopełniacz"  
Plural_Dative = "Liczba mnoga, Celownik"  
Plural_Accusative = "Liczba mnoga, Biernik"  
Plural_Instrumental = "Liczba mnoga, Narzędnik"  
Plural_Locative = "Liczba mnoga, Miejscownik"  
Plural_Vocative = "Liczba mnoga, Wołacz"
```

Klasa Verb(Enum)

Enumy dla czasownika

```
Infinitive = "Bezokolicznik"
Present_1_Singular = "Czas teraźniejszy, 1 osoba liczby pojedynczej"
Present_2_Singular = "Czas teraźniejszy, 2 osoba liczby pojedynczej"
Present_3_Singular = "Czas teraźniejszy, 3 osoba liczby pojedynczej"
Present_1_Plural = "Czas teraźniejszy, 1 osoba liczby mnogiej"
Present_2_Plural = "Czas teraźniejszy, 2 osoba liczby mnogiej"
Present_3_Plural = "Czas teraźniejszy, 3 osoba liczby mnogiej"
Imperative_2_Singular = "Tryb rozkazujący, 2 osoba liczby pojedynczej"
Imperative_1_Plural = "Tryb rozkazujący, 1 osoba liczby mnogiej"
Imperative_2_Plural = "Tryb rozkazujący, 2 osoba liczby mnogiej"
Past_1_Singular_Masculine = "Czas przeszły, 1 osoba liczby pojedynczej, rodzaj męski"
Past_2_Singular_Masculine = "Czas przeszły, 2 osoba liczby pojedynczej, rodzaj męski"
Past_3_Singular_Masculine = "Czas przeszły, 3 osoba liczby pojedynczej, rodzaj męski"
Past_1_Singular_Feminine = "Czas przeszły, 1 osoba liczby pojedynczej, rodzaj żeński"
Past_2_Singular_Feminine = "Czas przeszły, 2 osoba liczby pojedynczej, rodzaj żeński"
Past_3_Singular_Feminine = "Czas przeszły, 3 osoba liczby pojedynczej, rodzaj żeński"
Past_1_Singular_Neuter = "Czas przeszły, 1 osoba liczby pojedynczej, rodzaj nijaki"
Past_2_Singular_Neuter = "Czas przeszły, 2 osoba liczby pojedynczej, rodzaj nijaki"
Past_3_Singular_Neuter = "Czas przeszły, 3 osoba liczby pojedynczej, rodzaj nijaki"
Past_1_Plural_Masculine = "Czas przeszły, 1 osoba liczby mnogiej, rodzaj męskoosobowy"
Past_2_Plural_Masculine = "Czas przeszły, 2 osoba liczby mnogiej, rodzaj męskoosobowy"
Past_3_Plural_Masculine = "Czas przeszły, 3 osoba liczby mnogiej, rodzaj męskoosobowy"
Past_1_Plural_NonMasculine = "Czas przeszły, 1 osoba liczby mnogiej, rodzaj niemęskoosobowy"
Past_2_Plural_NonMasculine = "Czas przeszły, 2 osoba liczby mnogiej, rodzaj niemęskoosobowy"
Past_3_Plural_NonMasculine = "Czas przeszły, 3 osoba liczby mnogiej, rodzaj niemęskoosobowy"
Conditional_1_Singular_Masculine = "Tryb przypuszczający, 1 osoba liczby pojedynczej, rodzaj męski"
Conditional_2_Singular_Masculine = "Tryb przypuszczający, 2 osoba liczby pojedynczej, rodzaj męski"
Conditional_3_Singular_Masculine = "Tryb przypuszczający, 3 osoba liczby pojedynczej, rodzaj męski"
Conditional_1_Singular_Feminine = "Tryb przypuszczający, 1 osoba liczby pojedynczej, rodzaj żeński"
Conditional_2_Singular_Feminine = "Tryb przypuszczający, 2 osoba liczby pojedynczej, rodzaj żeński"
Conditional_3_Singular_Feminine = "Tryb przypuszczający, 3 osoba liczby pojedynczej, rodzaj żeński"
Conditional_1_Singular_Neuter = "Tryb przypuszczający, 1 osoba liczby pojedynczej, rodzaj nijaki"
Conditional_2_Singular_Neuter = "Tryb przypuszczający, 2 osoba liczby pojedynczej, rodzaj nijaki"
Conditional_3_Singular_Neuter = "Tryb przypuszczający, 3 osoba liczby pojedynczej, rodzaj nijaki"
Conditional_1_Plural_Masculine = "Tryb przypuszczający, 1 osoba liczby mnogiej, rodzaj męskoosobowy"
Conditional_2_Plural_Masculine = "Tryb przypuszczający, 2 osoba liczby mnogiej, rodzaj męskoosobowy"
Conditional_3_Plural_Masculine = "Tryb przypuszczający, 3 osoba liczby mnogiej, rodzaj męskoosobowy"
Conditional_1_Plural_NonMasculine = "Tryb przypuszczający, 1 osoba liczby mnogiej, rodzaj niemęskoosobowy"
Conditional_2_Plural_NonMasculine = "Tryb przypuszczający, 2 osoba liczby mnogiej, rodzaj niemęskoosobowy"
Conditional_3_Plural_NonMasculine = "Tryb przypuszczający, 3 osoba liczby mnogiej, rodzaj niemęskoosobowy"
Past_Impersonal = "Bezosobnik w czasie przeszłym"

Present_Adverbial_Part participle = "Imiesłów przysłówkowy współczesny"
Active_Adjectival_Part participle = "Imiesłów przymiotnikowy czynny"
Passive_Adjectival_Part participle = "Imiesłów przymiotnikowy bierny"
Perfect_Adverbial_Part participle = "Imiesłów przysłówkowy uprzedni"
Gerundive = "Rzeczownik odczasownikowy"
```

Klasa Adjective(Enum)

Enumy dla przymiotnika

```
Singular_Nominative_Masculine_Personal_Animate = "Liczba pojedyncza, Mianownik, rodzaj męski osobowy i męski żywotny"
Singular_Genitive_Masculine_Personal_Animate = "Liczba pojedyncza, Dopełniacz, rodzaj męski osobowy i męski żywotny"
Singular_Dative_Masculine_Personal_Animate = "Liczba pojedyncza, Celownik, rodzaj męski osobowy i męski żywotny"
Singular_Accusative_Masculine_Personal_Animate = "Liczba pojedyncza, Biernik, rodzaj męski osobowy i męski żywotny"
Singular_Instrumental_Masculine_Personal_Animate = "Liczba pojedyncza, Narzędnik, rodzaj męski osobowy i męski żywotny"
Singular_Locative_Masculine_Personal_Animate = "Liczba pojedyncza, Miejscownik, rodzaj męski osobowy i męski żywotny"
Singular_Vocative_Masculine_Personal_Animate = "Liczba pojedyncza, Wołacz, rodzaj męski osobowy i męski żywotny"
Singular_Nominative_Masculine_NonPersonal = "Liczba pojedyncza, Mianownik, rodzaj męski nieosobowy"
Singular_Genitive_Masculine_NonPersonal = "Liczba pojedyncza, Dopełniacz, rodzaj męski nieosobowy"
Singular_Dative_Masculine_NonPersonal = "Liczba pojedyncza, Celownik, rodzaj męski nieosobowy"
Singular_Accusative_Masculine_NonPersonal = "Liczba pojedyncza, Biernik, rodzaj męski nieosobowy"
Singular_Instrumental_Masculine_NonPersonal = "Liczba pojedyncza, Narzędnik, rodzaj męski nieosobowy"
Singular_Locative_Masculine_NonPersonal = "Liczba pojedyncza, Miejscownik, rodzaj męski nieosobowy"
Singular_Vocative_Masculine_NonPersonal = "Liczba pojedyncza, Wołacz, rodzaj męski nieosobowy"
Singular_Nominative_Feminine = "Liczba pojedyncza, Mianownik, rodzaj żeński"
Singular_Genitive_Feminine = "Liczba pojedyncza, Dopełniacz, rodzaj żeński"
Singular_Dative_Feminine = "Liczba pojedyncza, Celownik, rodzaj żeński"
Singular_Accusative_Feminine = "Liczba pojedyncza, Biernik, rodzaj żeński"
Singular_Instrumental_Feminine = "Liczba pojedyncza, Narzędnik, rodzaj żeński"
Singular_Locative_Feminine = "Liczba pojedyncza, Miejscownik, rodzaj żeński"
Singular_Vocative_Feminine = "Liczba pojedyncza, Wołacz, rodzaj żeński"
Singular_Nominative_Neuter = "Liczba pojedyncza, Mianownik, rodzaj nijaki"
Singular_Genitive_Neuter = "Liczba pojedyncza, Dopełniacz, rodzaj nijaki"
Singular_Dative_Neuter = "Liczba pojedyncza, Celownik, rodzaj nijaki"
Singular_Accusative_Neuter = "Liczba pojedyncza, Biernik, rodzaj nijaki"
Singular_Instrumental_Neuter = "Liczba pojedyncza, Narzędnik, rodzaj nijaki"
Singular_Locative_Neuter = "Liczba pojedyncza, Miejscownik, rodzaj nijaki"
Singular_Vocative_Neuter = "Liczba pojedyncza, Wołacz, rodzaj nijaki"
Plural_Nominative_Masculine = "Liczba mnoga, Mianownik, rodzaj męskoosobowy"
Plural_Genitive_Masculine = "Liczba mnoga, Dopełniacz, rodzaj męskoosobowy"
Plural_Dative_Masculine = "Liczba mnoga, Celownik, rodzaj męskoosobowy"
Plural_Accusative_Masculine = "Liczba mnoga, Biernik, rodzaj męskoosobowy"
Plural_Instrumental_Masculine = "Liczba mnoga, Narzędnik, rodzaj męskoosobowy"
Plural_Locative_Masculine = "Liczba mnoga, Miejscownik, rodzaj męskoosobowy"
Plural_Vocative_Masculine = "Liczba mnoga, Wołacz, rodzaj męskoosobowy"
Plural_Nominative_NonMasculine = "Liczba mnoga, Mianownik, rodzaj niemęskoosobowy"
Plural_Genitive_NonMasculine = "Liczba mnoga, Dopełniacz, rodzaj niemęskoosobowy"
Plural_Dative_NonMasculine = "Liczba mnoga, Celownik, rodzaj niemęskoosobowy"
Plural_Accusative_NonMasculine = "Liczba mnoga, Biernik, rodzaj niemęskoosobowy"
Plural_Instrumental_NonMasculine = "Liczba mnoga, Narzędnik, rodzaj niemęskoosobowy"
Plural_Locative_NonMasculine = "Liczba mnoga, Miejscownik, rodzaj niemęskoosobowy"
Plural_Vocative_NonMasculine = "Liczba mnoga, Wołacz, rodzaj niemęskoosobowy"
Positive_Form = "Stopień równy"
Comparative_Form = "Stopień wyższy"
Superlative_Form = "Stopień najwyższy"
```

Klasa Numeral(Enum)

Enumy dla liczebnika

```
Singular_Nominative_Masculine_Personal = "Liczba pojedyncza, Mianownik, rodzaj męski osobowy"
Singular_Genitive_Masculine_Personal = "Liczba pojedyncza, Dopełniacz, rodzaj męski osobowy"
Singular_Dative_Masculine_Personal = "Liczba pojedyncza, Celownik, rodzaj męski osobowy"
Singular_Accusative_Masculine_Personal = "Liczba pojedyncza, Biernik, rodzaj męski osobowy"
Singular_Instrumental_Masculine_Personal = "Liczba pojedyncza, Narzędnik, rodzaj męski osobowy"
Singular_Locative_Masculine_Personal = "Liczba pojedyncza, Miejscownik, rodzaj męski osobowy"
Singular_Vocative_Masculine_Personal = "Liczba pojedyncza, Wołacz, rodzaj męski osobowy"
Singular_Nominative_Masculine_Animate = "Liczba pojedyncza, Mianownik, rodzaj męski żywotny"
Singular_Genitive_Masculine_Animate = "Liczba pojedyncza, Dopełniacz, rodzaj męski żywotny"
Singular_Dative_Masculine_Animate = "Liczba pojedyncza, Celownik, rodzaj męski żywotny"
Singular_Accusative_Masculine_Animate = "Liczba pojedyncza, Biernik, rodzaj męski żywotny"
Singular_Instrumental_Masculine_Animate = "Liczba pojedyncza, Narzędnik, rodzaj męski żywotny"
Singular_Locative_Masculine_Animate = "Liczba pojedyncza, Miejscownik, rodzaj męski żywotny"
Singular_Vocative_Masculine_Animate = "Liczba pojedyncza, Wołacz, rodzaj męski żywotny"
Singular_Nominative_Masculine_NonPersonal = "Liczba pojedyncza, Mianownik, rodzaj męski nieosobowy"
Singular_Genitive_Masculine_NonPersonal = "Liczba pojedyncza, Dopełniacz, rodzaj męski nieosobowy"
Singular_Dative_Masculine_NonPersonal = "Liczba pojedyncza, Celownik, rodzaj męski nieosobowy"
Singular_Accusative_Masculine_NonPersonal = "Liczba pojedyncza, Biernik, rodzaj męski nieosobowy"
Singular_Instrumental_Masculine_NonPersonal = "Liczba pojedyncza, Narzędnik, rodzaj męski nieosobowy"
Singular_Locative_Masculine_NonPersonal = "Liczba pojedyncza, Miejscownik, rodzaj męski nieosobowy"
Singular_Vocative_Masculine_NonPersonal = "Liczba pojedyncza, Wołacz, rodzaj męski nieosobowy"
Singular_Nominative_Feminine = "Liczba pojedyncza, Mianownik, rodzaj żeński"
Singular_Genitive_Feminine = "Liczba pojedyncza, Dopełniacz, rodzaj żeński"
Singular_Dative_Feminine = "Liczba pojedyncza, Celownik, rodzaj żeński"
Singular_Accusative_Feminine = "Liczba pojedyncza, Biernik, rodzaj żeński"
Singular_Instrumental_Feminine = "Liczba pojedyncza, Narzędnik, rodzaj żeński"
Singular_Locative_Feminine = "Liczba pojedyncza, Miejscownik, rodzaj żeński"
Singular_Vocative_Feminine = "Liczba pojedyncza, Wołacz, rodzaj żeński"
Singular_Nominative_Neuter = "Liczba pojedyncza, Mianownik, rodzaj nijaki"
Singular_Genitive_Neuter = "Liczba pojedyncza, Dopełniacz, rodzaj nijaki"
Singular_Dative_Neuter = "Liczba pojedyncza, Celownik, rodzaj nijaki"
Singular_Accusative_Neuter = "Liczba pojedyncza, Biernik, rodzaj nijaki"
Singular_Instrumental_Neuter = "Liczba pojedyncza, Narzędnik, rodzaj nijaki"
Singular_Locative_Neuter = "Liczba pojedyncza, Miejscownik, rodzaj nijaki"
Singular_Vocative_Neuter = "Liczba pojedyncza, Wołacz, rodzaj nijaki"
Plural_Nominative_Masculine = "Liczba mnoga, Mianownik, rodzaj męskoosobowy"
Plural_Genitive_Masculine = "Liczba mnoga, Dopełniacz, rodzaj męskoosobowy"
Plural_Dative_Masculine = "Liczba mnoga, Celownik, rodzaj męskoosobowy"
Plural_Accusative_Masculine = "Liczba mnoga, Biernik, rodzaj męskoosobowy"
Plural_Instrumental_Masculine = "Liczba mnoga, Narzędnik, rodzaj męskoosobowy"
Plural_Locative_Masculine = "Liczba mnoga, Miejscownik, rodzaj męskoosobowy"
Plural_Vocative_Masculine = "Liczba mnoga, Wołacz, rodzaj męskoosobowy"
Plural_Nominative_NonMasculine = "Liczba mnoga, Mianownik, rodzaj niemęskoosobowy"
Plural_Genitive_NonMasculine = "Liczba mnoga, Dopełniacz, rodzaj niemęskoosobowy"
Plural_Dative_NonMasculine = "Liczba mnoga, Celownik, rodzaj niemęskoosobowy"
Plural_Accusative_NonMasculine = "Liczba mnoga, Biernik, rodzaj niemęskoosobowy"
Plural_Instrumental_NonMasculine = "Liczba mnoga, Narzędnik, rodzaj niemęskoosobowy"
Plural_Locative_NonMasculine = "Liczba mnoga, Miejscownik, rodzaj niemęskoosobowy"
Plural_Vocative_NonMasculine = "Liczba mnoga, Wołacz, rodzaj niemęskoosobowy"
```

Klasa Pronoun(Enum)

Enums dla zaimków

```
Singular_Nominative = "Liczba pojedyncza, Mianownik"  
Singular_Genitive = "Liczba pojedyncza, Dopełniacz"  
Singular_Dative = "Liczba pojedyncza, Celownik"  
Singular_Accusative = "Liczba pojedyncza, Biernik"  
Singular_Instrumental = "Liczba pojedyncza, Narzędnik"  
Singular_Locative = "Liczba pojedyncza, Miejscownik"  
Singular_Vocative = "Liczba pojedyncza, Wołacz"  
Plural_Nominative = "Liczba mnoga, Mianownik"  
Plural_Genitive = "Liczba mnoga, Dopełniacz"  
Plural_Dative = "Liczba mnoga, Celownik"  
Plural_Accusative = "Liczba mnoga, Biernik"  
Plural_Instrumental = "Liczba mnoga, Narzędnik"  
Plural_Locative = "Liczba mnoga, Miejscownik"  
Plural_Vocative = "Liczba mnoga, Wołacz"
```

Klasa Adverb(Enum)

Enums dla przysłówków

```
Positive_Form = "Stopień równy"  
Comparative_Form = "Stopień wyższy"  
Superlative_Form = "Stopień najwyższy"
```

Pozostałe

Dla Labels.UNINFLECTED, Labels.TEXT, Labels.ACRONYM nie ma specjalnych enumów ponieważ nie ma takiej potrzeby – są one nieodmienne.