

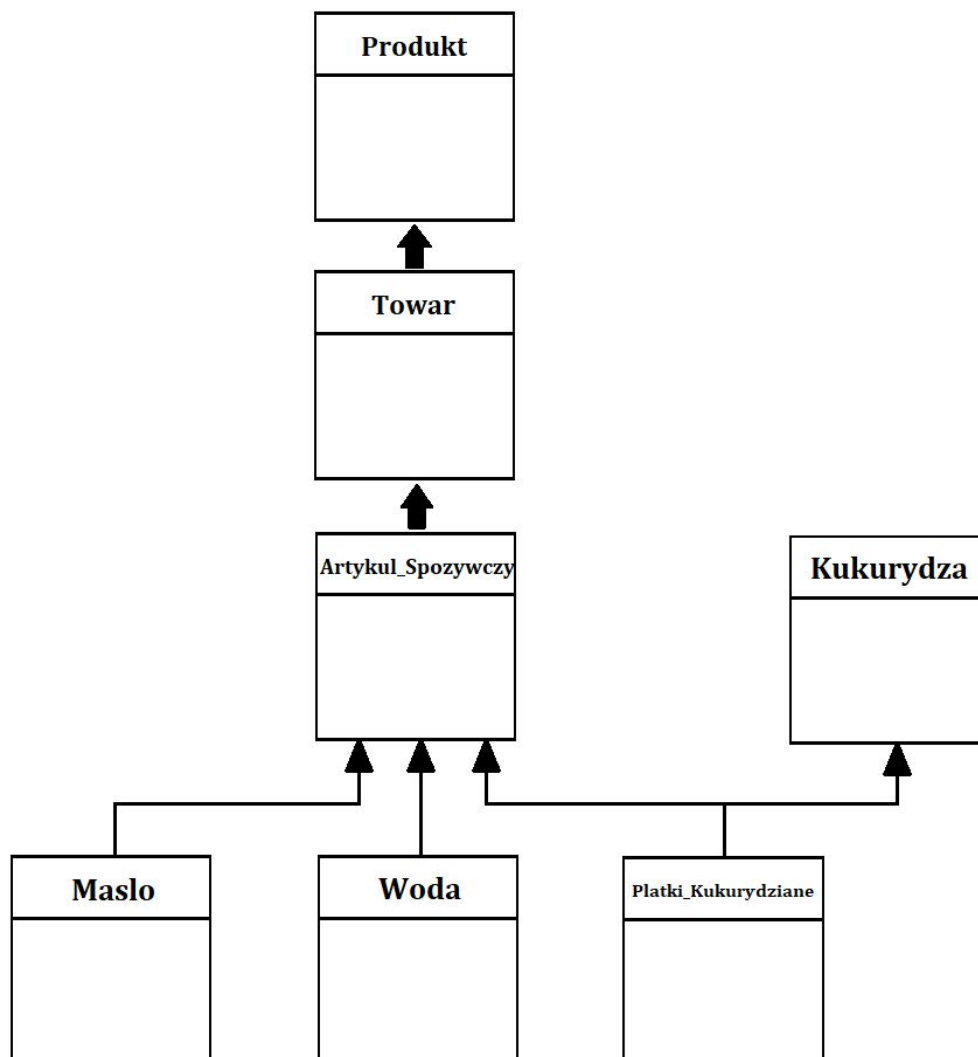
Mikołaj Ryba

Inżynieria Danych

ARCHITEKTURA KLAS

Oczywiście kukurydza może należeć do klasy towar ale wolałem nie tworzyć romba dziedziczenia o którym pan nas ostrzegał.

Nie wpisywałem zmiennych i metod do blozków poniżej bo wyglądało to nieestetycznie.



INFORMACJE DOTYCZĄCE KODU

Pana klase do dat prawie nie ruszałem.

Komentarze i docstringi są napisane językiem pseudo polsko-angielsko-pythonowym stylem numpydoc.

Mogłem dodać stringi lub docstringi do method ale uznałem że jest kompletnie niepotrzebne przy tak banalnym kodzie

Rozszerzałem doctsringi child klass przez docstringi parentklass co nie wiem czy jest poprawne.

Próbowałem użyć zmiennych niepublicznych(____) wszędzie ale pojawiły się pewne komplikacje podczas dziedziczenia. Więc użyłem jednego underscorea(podłogi) u większości zmiennych co jest konwencją w pythonie że można coś dynamicznie zmienić ale lepiej tego nie robić.

Troche inaczej podpiąłem tą klase do tworzenia dat. Zamiast używać jej dynamicznie poza klasą, użyłem jej w Klasie.

@property sprawia że funkcje możemy wezwać jak attribut. Użyłem tego dekoratora przy tych metodach gdzie nie jest wymagane podanie parametrów

INFORMACJE DOTYCZĄCE DOKUMENTACJI KLASS

Nie za bardzo wiedziałem jak stworzyć dokumentacje więc, uznałem że te zmienne przy minusie(-) są zmiennymi obiektu stworzonymi w konstruktorze.

Plusy to metody których można użyć (wraz setterami i getterami). W każdej klasie są zrobione setery i gettery (z 30) do wszystkich attributów, jedynie nie uwzględniłem ich w dokumentacji poniżej.

Rozszerzenie poniższej dokumentacji znajduje się w docstringach klasy które można wezwać poprzez

class.__doc__
lub
help(class) .

Data

```
- self.dni : ints  
+ __init__(self, ...) : void  
+ __str__() : str  
+ __sub__(other) : int  
+ setDzien(d) : void  
+ setMiesiac(m) : void  
+ setRok(r) : void  
+ setData(data_napis) : void  
+ getDzien() : int  
+ getMiesiac() : int  
+ getRok() : int  
+ getData() : tuple  
+ getDniOd1900() : int  
+ jutro() : void  
+ wczoraj() : void  
+ wPrzyszlosc(n) : void  
+ wPrzeszlosc(n) : void  
+ porownajDaty(data1) : int  
+ roznicaDat(data1) : int
```

Produkt

```
- self._nazwa : str
- self._producent : str
- self._data_produkcji : tuple

+ __init__(self, ...) : void
+ __str__() : str

+ posiadane_produkty : str
+ reset_posiadane_produkty(password) : void
+ create_email() : void
+ email : str <-getter
+ email : void <-setter
+ email : void <-deleter
```

Towar

```
- self._nr_Inwent : int
- self._data_dostawy : tuple
- self._cena : float
- self._ilosc : int

+ __init__(self, ...) : void
+ __str__() : str

+ sprzedaj(ile) : str
+ ilosc_towarow_na_magazynie : int
+ wartosc_towarow_na_magazynie : float
+ wartosc_danego_towaru : float
```

Artykul_Spozywczy

```
- self._termin_przydatnosci  
- self._opakowanie : str  
  
+ __init__(self, ...) : void  
+ __str__() : str  
  
+ dni_do_przeterminowania(d, m, r) : int/str
```

Kukurydza

```
- self._odmiana_kukurydzy : str  
- self._data_zasiania : tuple  
- self._data_zbioru : tuple  
- self._ekologiczne : bool  
  
+ __init__(self, ...) : void  
+ __str__() : str  
  
+ place_dla_rolnika_stasia : int
```

Platki_Kukurydziane

```
- self._pojemnosc_gramy  
  
+ __init__(self, ...) : void  
+ __str__() : str  
  
+ wartosci_odzywcze : str
```

Maslo

```
- self._waga : float
- self._rodzaj : str

+ __init__(self, ...) : void
+ __str__() : str

+ wartosc_odzywcza : str
+ maslo_info : str
+ ktore_zdrowsze(other) : str
```

Woda

```
- self._rodzaj : str
- self._gazowana : tuple
- self._pojemnosc_ml : float

+ __init__(self, ...) : void
+ __str__() : str
+ __lt__() : bool      (self._pojemnosc_ml) comparison
+ __le__() : bool      (self._pojemnosc_ml) comparison
+ __eq__() : bool      (self._pojemnosc_ml) comparison
+ __ne__() : bool      (self._pojemnosc_ml) comparison
+ __gt__() : bool      (self._pojemnosc_ml) comparison
+ __ge__() : bool      (self._pojemnosc_ml) comparison

+ woda_info : str
```