Wizualizacja danych Dr Anna Muranova

Semestr letni 2024, UWM w Olsztynie

5. Zajęcie (03-04.04.2024)

Pierwszym krokiem tworzenia własnych obiektów jest stworzenie klasy, która jest czymś w rodzaju projektu, na podstawie którego tworzone są obiekty, które są egzemplarzami klasy. Obiekty mogą przechowywać dane, a także wykonywać kod, który jest umieszczony w metodach. Metody są funkcjami, które są częściami klasy.

Klasy można umieszczać w osobnych plikach (modułach), zwłaszcza jeśli są bardzo rozbudowane, ale wiele klas może być także umieszczonych w jednym module.

Przy nazywaniu klas możemy używać tych samych znaków co w przypadku zmiennych i funkcji: małych i dużych liter oraz znaku podkreślenia `_`. Konwencja nazewnicza jest jednak inna. Dla funkcji i zmiennych używaliśmy `snake_case`. W przypadku klas używamy `PascalCase` (lub `UpperCamelCase`), czyli każde słowo zaczynamy wielką literą, nie używamy spacji ani znaków podkreślenia.

`https://www.w3schools.com/python/python_classes.asp`
`https://kt.academy/pl/article/py-klasy`
`https://ggoralski.gitlab.io/python-wprowadzenie/czesc_i/15-klasy_i_obiekty/`

Ćwiczenie 1. Stwórz program a w nim klase `Car`, w której będzie można przechowywać markę samochodu i rok produkcji (zmienne marka i rok). Stwórz nowy obiekt klasy `Car` jako `car1`, ustaw dla niego dowolna markę i rok produkcji. Stwórz drugi obiekt `car2` typu `Car`, przypisz mu dowolne wartości i wyświetl na ekranie. Przypisz do siebie wartości `car1` i **car2** (`car1 = car2`) i ponownie sprawdź na ekranie jakie wartości otrzymasz.

Ćwiczenie 2. Klas Owoce:

```
class Fruit:
def __init__(self, color, weight):
        self.color = color
        self.weight = weight
```

- Stwórz klasy potomne `Apple`, `Banana`, `Orange` z domyślą inicializację.
- Dodaj funkcje `isfresh()` (w zależności od koloru zwraca `True` albo `False`).

Poćwicz różny operacje z tymi klasami (naprz. wyświetlenie na konsoli)

Ćwiczenie 3.
- Stwórz klasę `Account` z kilkoma polami, metodami i konstruktorami (saldo początkowe, końcowe, przelew miedzy kontami, przelew zewnętrzny, wplata, wyplata itd.) Następnie utwórz co najmniej 3 obiekty i wykonaj kilka metody na nich.

- Rozbuduj kod i stwórz klasy potomne `PrivatAccount` (rozbuduj o np. przelew wynagrodzenia, itp) i `FirmAccount` (rozbudowany o np. przelewy do ZUS, US).

Ćwiczenie 4. Stwórz klasę `Romanian` liczb rzymskich. Dodaj metody $+, -, *$ oraz możliwość wyświetlania na konsoli przy pomocy `print` (dodatkowo można dodać: `len`,//, możliwość indeksacji, ...).

`https://pl.wikipedia.org/wiki/Rzymski_system_zapisywania_liczb`

W liczbach rzymskich nie ma 0.

Jak teraz pracować z tą klasą na konsoli?