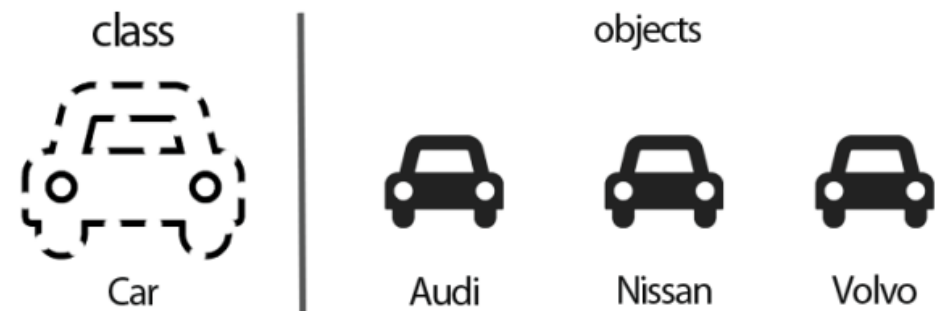


Podstawy OOP

Podstawy programowania w języku Python



Programowanie proceduralne

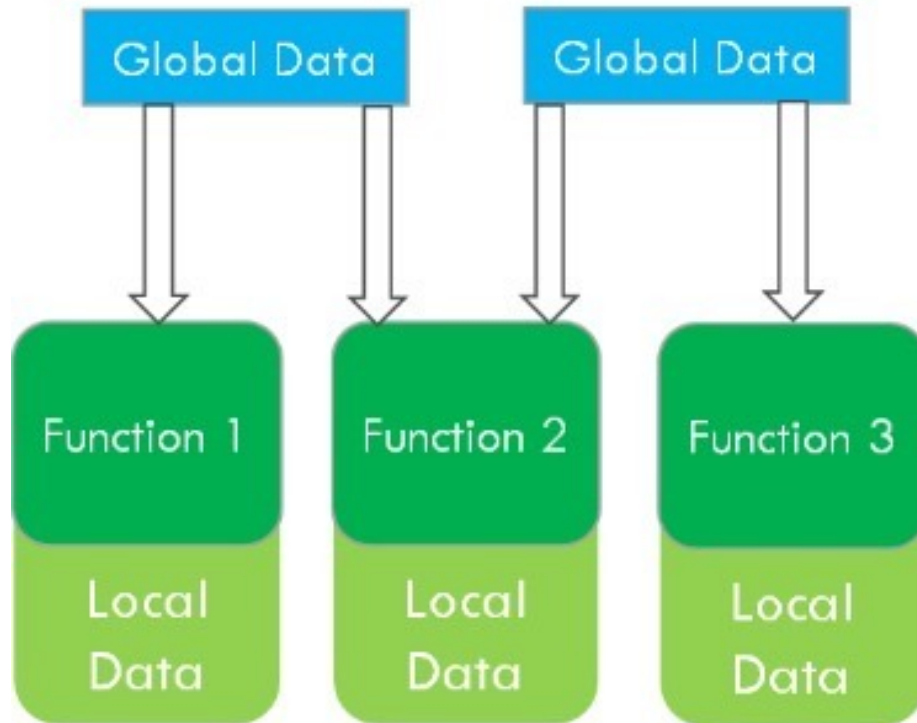
- styl programowania (paradygmat)
- dzielenie kodu na procedury, czyli fragmenty wykonujące ściśle określone operacje
- dominujące podejście do tworzenia oprogramowania przez dziesięciolecia - jest nadal w użyciu
- programista pisze sekwencję czynności wykonywanych w celu dotarcia do rozwiązania - każda z tych czynności zmienia stan programu

Programowanie obiektowe

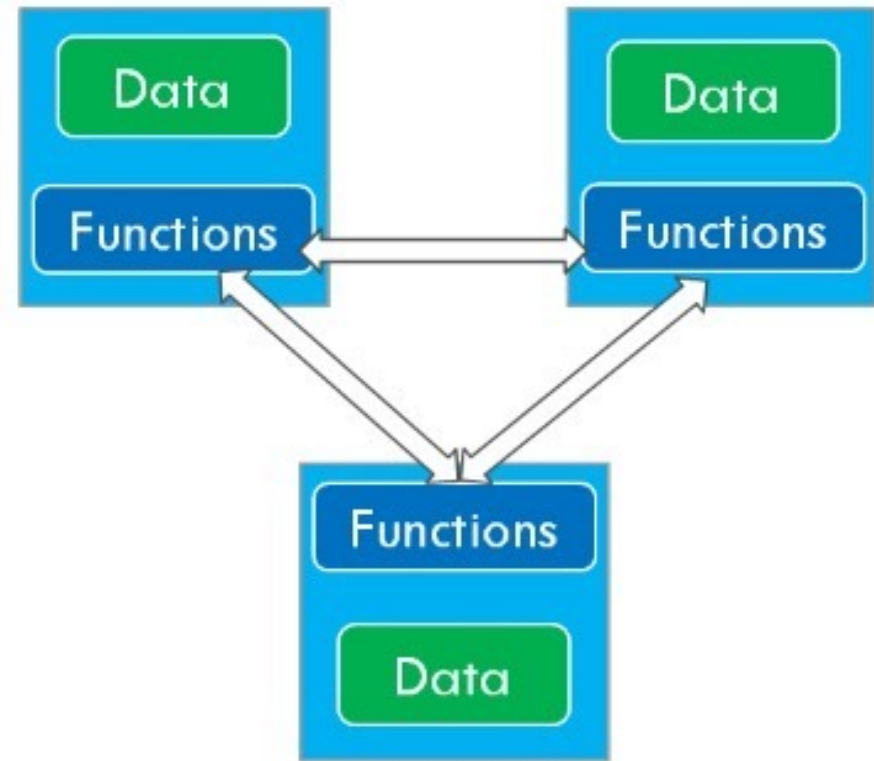
- styl programowania (paradygmat)
- programowania polegający na tworzeniu obiektów, które prowadzą ze sobą interakcje
- podejście szczególnie przydatne w przypadku dużych i złożonych projektów realizowanych przez duże zespoły składające się z wielu programistów

POP vs. OOP

Procedural Oriented Programming

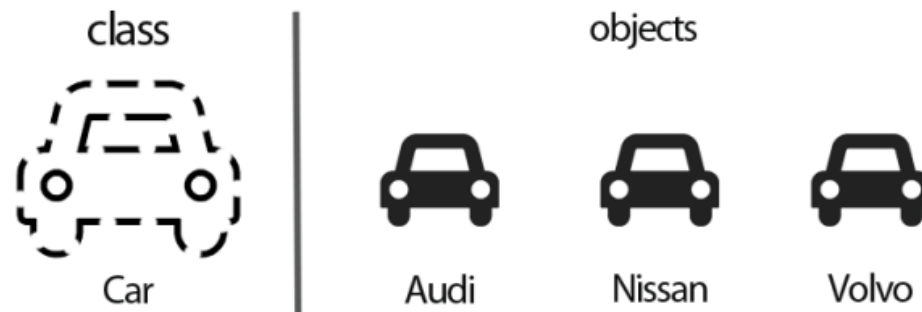


Object Oriented Programming



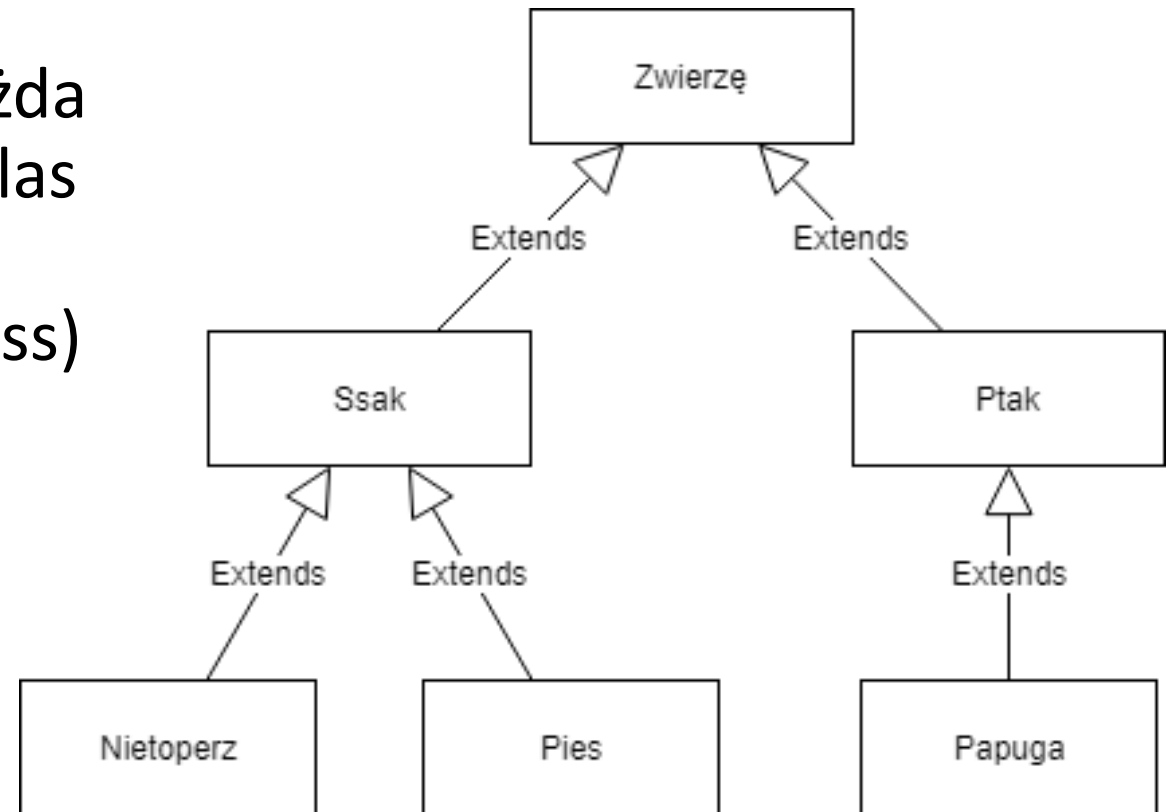
Klasy i obiekty

- klasy i obiekty są fundamentami programowania zorientowanego obiektowo (OOP)
- klasa to szablon, który definiuje cechy i zachowania obiektu
- obiekt to instancja klasy, czyli konkretny przykład obiektu



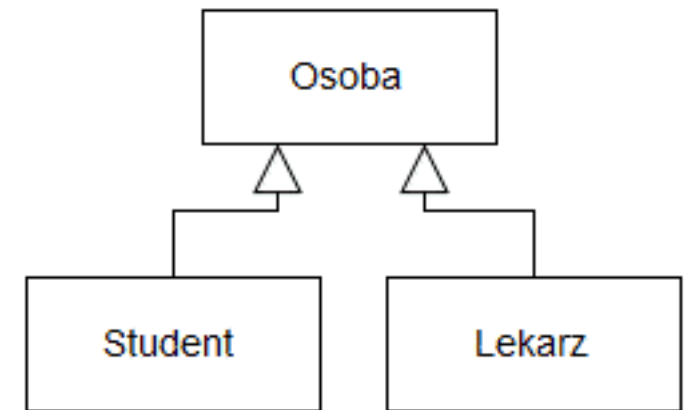
Hierarchia klas

- drzewo dziedziczenia, w którym każda klasa może mieć jedną lub więcej klas nadrzędnych (ang. superclass) oraz wiele klas pochodnych (ang. subclass)
- klasa pochodna dziedziczy cechy i metody z klasy nadrzędnej



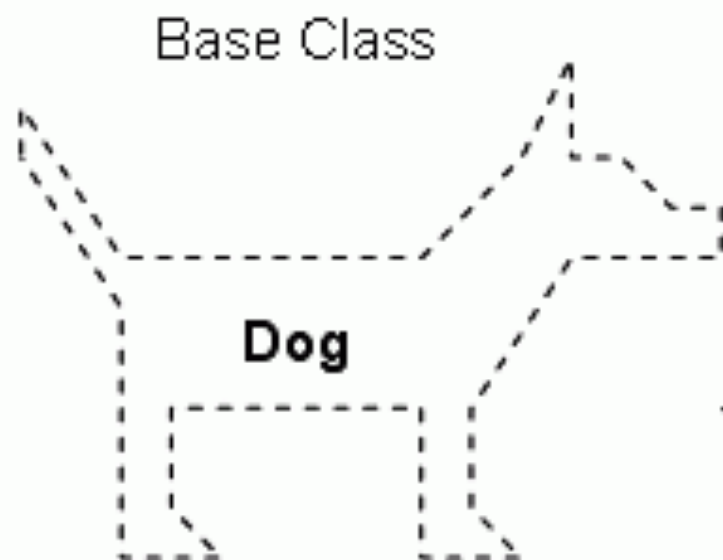
Dziedziczenie

- jedna z podstawowych koncepcji programowania obiektowego
- umożliwia tworzenie nowych klas poprzez dziedziczenie cech (atrybutów) i zachowań (metod) z klas istniejących
- każdy obiekt powiązany z określonym poziomem hierarchii klas dziedziczy wszystkie cechy zdefiniowane w dowolnej z nadklas

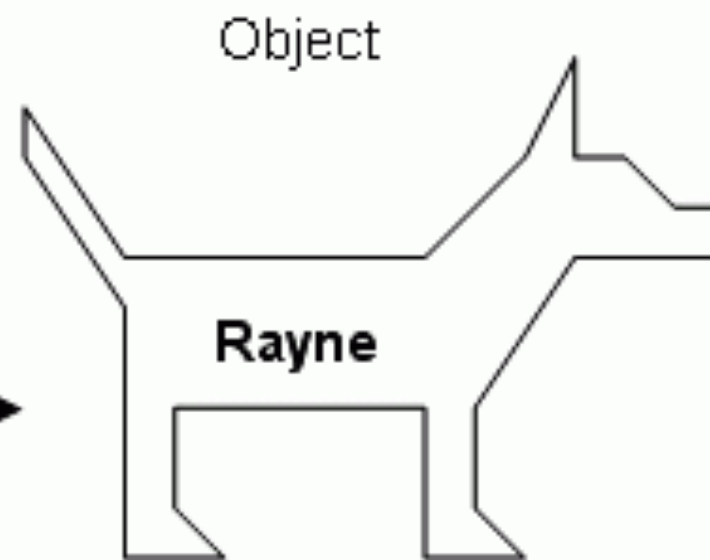


Co zawiera obiekt?

- **nazwę**, która jednoznacznie identyfikuje go w macierzystej przestrzeni nazw (choć mogą istnieć również anonimowe obiekty)
- obiekt ma **zestaw indywidualnych właściwości**, które czynią go oryginalnym, unikalnym lub wybitnym (choć możliwe jest, że niektóre obiekty nie będą mieć żadnych właściwości)
- obiekt ma zestaw **umiejętności do wykonywania określonych czynności**, zdolnych zmienić sam obiekt lub niektóre inne obiekty



Create
Instance



Properties

Color
Eye Color
Height
Length
Weight

Methods

Sit
Lay Down
Shake
Come

Property values

Color: Gray, White, and Black
Eye Color: Blue and Brown
Height: 18 Inches
Length: 36 Inches
Weight: 30 Pounds

Methods

Sit
Lay Down
Shake
Come

Tworzenie klasy i obiektu

- aby zdefiniować klasę Pythona, należy użyć słowa kluczowego **class**

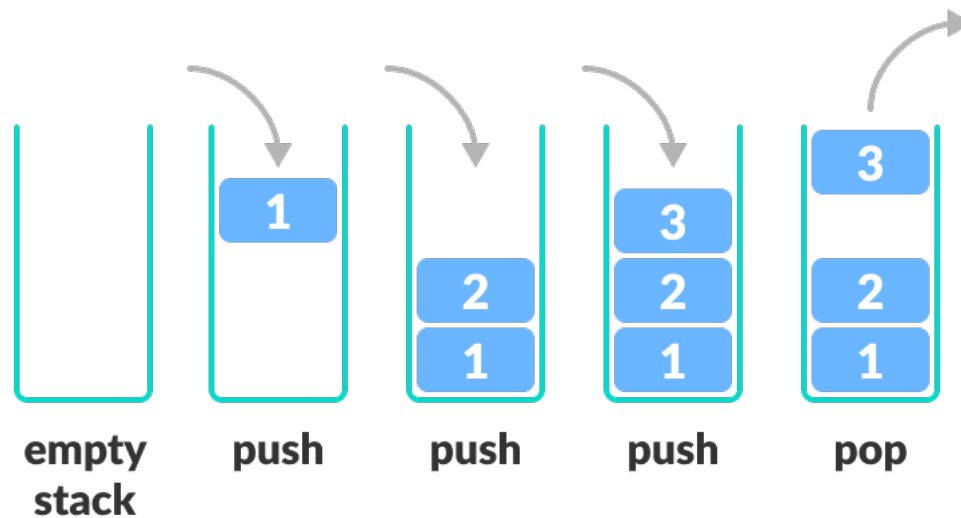
```
1 class MyClass:  
2     pass
```

- aby stworzyć obiekt wcześniej zdefiniowanej klasy, należy użyć tej klasy tak, jakby była funkcją

```
1 my_object = MyClass()
```

Stos (stack)

- struktura danych, która umożliwia przechowywanie elementów w sposób "last-in, first-out" (LIFO)
- ideę stosu danych można zilustrować jako stos położonych jedna na drugiej książek



Pytanie

Jakie są zalety programowania obiektowego w Pythonie?

- a) łatwiejsza organizacja kodu, łatwiejsze przystosowanie do zmian, łatwiejsze testowanie i debugowanie
- b) szybsze wykonywanie programu, łatwiejsza optymalizacja kodu, mniejsze zużycie pamięci
- c) bardziej elastyczne i niestandardowe rozwiązania, łatwiejsza komunikacja między różnymi językami programowania
- d) bardziej intuicyjne i zrozumiałe programowanie dla początkujących

Odpowiedź: a)

Pytanie

Czym jest klasa w programowaniu obiektowym?

- a) instancją klasy, czyli konkretnym wystąpieniem tej klasy w programie
- b) zmienną, która przechowuje dane związane z obiektem, takie jak nazwa, wartość, rozmiar itp.
- c) szablonem lub wzorcem, który opisuje cechy i zachowania obiektów tworzonych na jej podstawie
- d) funkcją, która pozwala na interakcję z obiektem i wykonywanie określonych działań na jego atrybutach

Odpowiedź: c)

Pytanie

Wskaż błędne stwierdzenie:

- a) obiekt posiada nazwę
- b) klasa może dziedziczyć zachowanie po nadklasie
- c) na podstawie obiektu można wygenerować kilka klas
- d) w klasie definiujemy atrybuty oraz metody
- e) obiekt to instancja klasy

Odpowiedź: c)

Na podstawie klasy generujemy obiekty (egzemplarze) a nie odwrotnie.