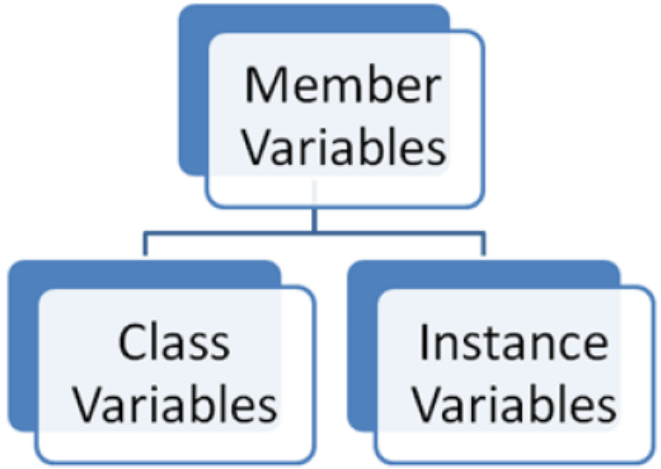
OOP: Właściwości

Podstawy programowania w języku Python



Zmienne instancji

- należą do poszczególnych obiektów danej klasy
- każdy obiekt posiada własną kopię takiej zmiennej
- nie są one dzielone ani w żaden sposób powiązane ze sobą w różnych instancjach danej klasy

```
1  class MyClass:
2      def __init__(self):
3          self.x = 1  # zmienna instancji
4
5  obj = MyClass()
6  print(obj.x)  # 1
```

Zmienne klasy

- są dostępne dla wszystkich instancji danej klasy
- istnieje tylko jedna kopia zmiennej klasy
- jeśli jeden obiekt zmieni w jakiś sposób tę zmienną, to zmiana ta będzie widziana również przez wszystkie pozostałe instancje

```
1  class MyClass:
2      y = 7 # zmienna klasy
3
4  print(MyClass.y) # 7
5
6  obj = MyClass()
7  print(obj.y)  # 7
```

Właściwość __dict__

- predefiniowana właściwość każdego obiektu i klasy
- słownik zawierający nazwy i wartości wszystkich atrybutów, które obecnie posiada obiekt (zmiennych instancji) lub klasa (zmiennych klasy i metod)

```
1  class MyClass:
2      y = 7 # zmienna klasy
3      def __init__(self):
4          self.x = 1 # zmienna instancji
5
6  print(MyClass.__dict__)
7
8  obj = MyClass()
9  print(obj.__dict__)
```

Zmienne niepubliczne

- zmienną niepubliczną jest zmienna poprzedzona dwoma podkreśleniami i zakończona co najwyżej jednym podkreśleniem
- dostęp jedynie z poziomu danej klasy
- zapobiegają użyciu zmiennych prywatnych poza klasą (enkapsulacja)
- pozwala uniknąć kolizji nazw zmiennych podczas dziedziczenia

do zmienne prywatnej można się jednak odwołać za pomocą poniższego schematu _NazwaKlasy__nazwaZmiennejPrywatnej

```
1   class MyClass:
2       def __init__(self):
3           self.__x = 1 # prywatna zmienna instancji
4
5       def test(self):
6           return self.__x # dostęp z poziomu klasy
7
8   obj = MyClass()
9   #print(obj.__x) # wyjątek AttributeError
10  print(obj.test())
11  print(obj._MyClass__x)
```

Sprawdzanie czy atrybut istnieje

- funkcja **hasattr()** potrafi bezpiecznie sprawdzić, czy dany obiekt/klasa zawiera określony atrybut (zmienną instancji lub metodę)
- pobiera:
  - klasę lub obiekt podlegający sprawdzeniu
  - nazwę atrybutu, której której istnienie należy potwierdzić lub zanegować (uwaga: musi to być ciąg znaków zawierający nazwę atrybutu, a nie sama nazwa)
- zwraca True lub False

```
1    if hassattr(obj, "x"):
2        print(obj.x)
```

Pytanie

Jeżeli chcemy zdefiniować zmienną w klasie, dzięki której każdy obiekt będzie przechowywał własną kopię tej zmiennej to będziemy musieli użyć:

a) zmiennej klasy
b) zmiennej instancji
c) zmiennej lokalnej
d) żadnej z powyższych

Odpowiedź: b)
Zmienne instancji to zmienne, które nie są dzielone ani w żadnej sposób powiązane ze sobą w różnych instancjach danej klasy.

Pytanie

Co wyświetli się na ekranie po wykonaniu poniższego skryptu?

a) None
b) 5
c) Nie ma takiej właściwości
d) wystąpi błąd AttributeError

```python
class Python:
    __x = 5

if hasattr(Python, "__x"):
    print(Python.__x)
else:
    print("Nie ma takiej właściwości")
```

Odpowiedź: c)

Pole __x jest polem niepublicznym i nie jest dostępne (pod taką) nazwą poza klasą.

Pytanie

Jak poprawnie odwołać się do zmiennej __x poniższej klasy, poza nią?

a) obj.__x
b) self.__x
c) Python._Python__x
d) self._Python__x
e) Python.__x

```python
class Python:
    __x = 5

obj = Python()
# tutaj odwołanie do __x
```

Odpowiedź: c)
Jest to zmienna klasy, możemy się do niej odwołać za pomocą nazwy klasy, ale ze względu na jej niepubliczność, należy użyć schematu _NazwaKlasy__nazwaZmiennej.