



Pystart.pl

Statycznie: metody i właściwości

lekcja czterdziesta trzecia

PyStart #43 Statycznie i z klasą

Metody statyczne



```
1 class Phone:
2     def __init__(self, country, number):
3         self.country = country
4         self.number = number
5
6     @staticmethod
7     def get_code(code: str):
8         codes = {
9             'pl': 48,
10            'de': 49,
11            'ru': 7
12        }
13
14        return codes[code]
15
16     def get_full_number(self):
17         return self.get_code(self.country) + self.number
18
```

20

`Phone.get_code('pl')`

Nie ma kontekstu obiektu, nie można używać `self`. Po prostu “grupujemy” funkcje do postaci metod.

PyStart #43 Statycznie i z klasą

Inny przykład

```
1  class App:
2      @staticmethod
3      def main():
4          print('hello world')
5
6      @staticmethod
7      def get_name():
8          return 'Application'
9
10
11  if __name__ == '__main__':
12      App.main()
13
14      app = App
15      print(app.get_name())
16
```

- `main()` nie potrzebuje nowej instancji klasy `App`
- metody statyczne wywołam jednak też na obiekcie (linia 15)

PyStart #43 Statycznie i z klasą

Właściwości statyczne

NEXT_ID = 1

+1



id **1**
marka **Lada**
model **Samara**
przebieg **100**

NEXT_ID = 2

+1



id **2**
marka **Renault**
model **Clio**
przebieg **15**

NEXT_ID = 3

id **3**
marka ...
model ...
przebieg ...

PyStart #43 Statycznie i z klasą

Właściwości statyczne



```
1 class Student:
2     NEXT_ID = 1
3
4     def __init__(self, first_name, last_name):
5         self.id = Student.NEXT_ID
6         self.first_name = first_name
7         self.last_name = last_name
8         Student.NEXT_ID += 1
9
10    def hello(self):
11        return f'{self.id}. {self.first_name} {self.last_name}'
12
```

→ Właściwość statyczna jest współdzielona między wszystkie obiekty.

PyStart #43 Statycznie i z klasą

Inny przykład - singleton



```
1  class Singleton:
2      _INSTANCE = None
3
4      @staticmethod
5      def get_instance():
6          if Singleton._INSTANCE is None:
7              Singleton._INSTANCE = Singleton()
8
9          return Singleton._INSTANCE
10
11
12  app = Singleton.get_instance()
13  print(app)
14
```

→ Za każdym razem zwracany jest ten sam obiekt.

PyStart #43 Statycznie i z klasą

Zadania dla nabrania wprawy

Przygotuj klasę File, która będzie posiadała następujące metody statyczne

- **delete(filename:str)** - usunięcie konkretnego pliku
- **read_file(filename:str)** - pobranie listy zawierającej wszystkie linie pliku
- **exists(filename: str)** - zwróci True/False w zależności od tego czy plik istnieje

Klasa mogłaby posiadać oczywiście także inne funkcjonalności dotyczące operacji na plikach, zwróć jednak uwagę, że stworzone powyżej metody będą też przydatne poza obiekowym aspektem.

43.1

