



Pystart.pl

Metody klasowe

lekcja czterdziesta czwarta

PyStart #44 Metody klasowe

Wprowadzenie

→ Czasem jeden init to za mało, gdyż:

- ◆ mamy zbyt wiele argumentów, a chcemy uprościć tworzenie wyspecjalizowanych obiektów
- ◆ trzeba na obiekcie wykonać kilka dodatkowych metod aby stworzyć go w pełni funkcjonalnym



PyStart #44 Metody klasowe

Wprowadzenie

```
1 class Airplane:
2     def __init__(self, model, engines=2):
3         self.model = model
4         self.engines = engines
5
6     @classmethod
7     def get_ready(cls, model):
8         sample_plane = cls(model, 3)
9         sample_plane.engine_on(0)
10        sample_plane.engine_on(1)
11        sample_plane.engine_on(2)
12
13        return sample_plane
14
```



PyStart #44 Metody klasowe

Wprowadzenie

```
1 class Airplane:
2     def __init__(self, model, engines):
3         self.model = model
4         self.engines = engines
5
6     @classmethod
7     def get_ready(cls, model):
8         sample_plane = cls(model, 3)
9         sample_plane.engine_on(0)
10        sample_plane.engine_on(1)
11        sample_plane.engine_on(2)
12
13        return sample_plane
14
19 airplane = Airplane('ABC', 3)
20 airplane.engine_on(0)
21 airplane.engine_on(1)
22 airplane.engine_on(2)
23
24 airplane = Airplane.get_ready('ABC')
```



PyStart #44 Metody klasowe

Zadania dla nabrania wprawy

Dana jest klasa **Person**, która w konstruktorze odbiera imię i nazwisko. Bardzo często jednak obiekty tej klasy tworzone są także z pliku CSV, którego wiersze wyglądają tak:

Jan;Kowalski

Zofia;Nowak

Krystyna;Krychowiak

przygotuj metodę klasową o nazwie **from_row**, która będzie tworzyła nam obiekty klasy **Person** wygodnie.

44.1



PyStart #44 Metody klasowe

Zadania dla nabrania wprawy

Dana jest klasa **Item**, która posiada dekorator **@dataclass** i pozwala na tworzenie obiektów z właściwościami takimi jak: **name(str)**, **price(float)**, **discount(float)** (<https://docs.python.org/3/library/dataclasses.html>)

oraz klasa **Collection**, która pozwala na dodawanie do niej obiektów:

```
collection = Collection()
collection.items.append(item)
```

przygotuj metodę klasową, która umożliwi stworzenie kolekcji wraz z elementami w taki sposób: **Collection.create_with_items(item1, item2, item3)**

44.2

