**Programowanie obiektowe**

# PRZED ZAJĘCIAMI

1. Zapoznaj się z zasadami i technikami programowania obiektowego w Pythonie — lekcje od 54 do 59 w „Wstępie do informatyki i programowania w języku Python (MIT)”.

[https://youtube.com/playlist?list=PLRJdqdXieSHN0U9AdnmwD-9QcR9hmw04d](https://translate.google.com/translate?hl=pl&prev=_t&sl=en&tl=pl&u=https://youtube.com/playlist%3Flist%3DPLRJdqdXieSHN0U9AdnmwD-9QcR9hmw04d)

1. Zapoznaj się z zasobami dostępnymi w w3schools na temat dziedziczenia i tworzenia klas pochodnych:

[https://www.w3schools.com/python/python\_inheritance.asp](https://translate.google.com/translate?hl=pl&prev=_t&sl=en&tl=pl&u=https://www.w3schools.com/python/python_inheritance.asp)

1. Zapoznaj się z regułami dziedziczenia w samouczku Pythona, rozdział 9.5:

[https://docs.python.org/3/tutorial/index.html](https://translate.google.com/translate?hl=pl&prev=_t&sl=en&tl=pl&u=https://docs.python.org/3/tutorial/index.html)

# PODCZAS LEKCJI

## Ciąg reprezentacja obiektu

1. Dla wygody i czytelności kodu programu istnieje możliwość stworzenia tekstowej reprezentacji obiektu w postaci ciągu. Taki obiekt może być następnie użyty wszędzie tam, gdzie wymagane są dane łańcuchowe, np. podczas wywoływania print().

Uruchom poniższy program. Zwróć uwagę na metodę \_\_str\_\_ i wywołanie funkcji print().

class University(): def \_\_init\_\_(self, name): self.name = name def \_\_str\_\_(self): return self.name + "jest najlepszy!" my\_university = Uniwersytet('UEK Kraków') print(my\_university)  
     
     
         
     
     
         
   

1. Utwórz klasę, która reprezentuje utwory muzyczne. Zdefiniuj konstruktor klasy, który pozwala ustawić początkowe wartości utworu muzycznego (wykonawca, tytuł utworu, album, rok) podczas tworzenia obiektu. Uzupełnij klasę metodą \_\_str\_\_, zwracając dane utworu jako ciąg znaków w formacie jak poniżej (4 wiersze).

Wykonawca: Ed Sheeran  
Piosenka: Hearts Don't Break Around Here Album: Divide Rok: 2017        
    

Następnie utwórz trzy obiekty, które reprezentują trzy różne utwory muzyczne. Wyświetl te obiekty.

## Dziedzictwo

1. W programowaniu obiektowym dziedziczenie jest mechanizmem współdzielenia funkcjonalności między klasami. Klasa pochodna (dziedzicząca), stworzona na bazie klasy bazowej, posiada własne cechy i zachowania, a dodatkowo czerpie (dziedziczy) cechy i zachowania również z klasy bazowej.

Uruchom poniższy program. Wtedy odpowiedz na pytania:

1. Jak ustala się dziedziczenie
2. Która klasa jest klasą bazową, a która klasą pochodną
3. Ile funkcji i zachowań ma klasa bazowa
4. Ile własnych cech i zachowań posiada klasa pochodna
5. Jakie cechy i zachowania klasa pochodna dziedziczy po klasie bazowej?
6. Jak wywołać konstruktora klasy bazowej, aby zainicjować pola w klasie bazowej?
7. Jak odwoływać się do pól klasy bazowej z klasy pochodnej?
8. Jak odwoływać się do metod klasy bazowej z klasy pochodnej?

class Osoba(): def \_\_init\_\_(self,name): self.name = imię def pozdrowienia(self): print(f'Witam wszystkich! Jestem {self.name}') class Nauczyciel(Person): def \_\_init\_\_ (siebie,imię,uczelnia): self.university = uczelnia super().\_\_init\_\_(imię) def say(self): print(f'Pracuję jako nauczyciel na {self.university}') def bye(self): print(f'A teraz {self.name} żegna się!') t = Nauczyciel('Johnny','UEK') t.greet() t.say() t.bye()  
     
         
     
         
  
  
     
         
         
     
         
     
         
       

1. Poniższa klasa opisuje każdą wiadomość tekstową.

class Message(): def \_\_init\_\_(self): self.message = '' def set\_message(self,message): pass  
     
         
   

Uzupełnij definicję metody set\_message() tak, aby podczas ustawiania treści wiadomości jej pierwsza litera była zmieniana na wielką, a reszta na małą. Dodaj także BYE na końcu wiadomości.

Następnie utwórz dwie klasy pochodne, każdą w osobnym pliku: SMS i Email. Klasa SMS powinna zawierać pole opisujące numer telefonu nadawcy i odbiorcy, natomiast klasa Email powinna zawierać pola: adres nadawcy, adres odbiorcy, temat wiadomości e-mail. W obu klasach zdefiniuj metodę send() do wysłania wiadomości. Podczas wysyłania wyświetlaj dane dla danego typu wiadomości, np.:

Wysyłam e-mail...  
Od: nowak@onet.pl Do: kowalski@gmail.com Temat: Spotkanie w czwartek Uprzejmie informujemy, że nasze czwartkowe spotkanie zostało odwołane. PA.     
     

Stwórz program, w którym wyślesz jeden e-mail i jeden SMS.

# PO ZAJĘCIACH

1. Utwórz klasę opisującą telefony komórkowe z co najmniej 3 stanami telefonu i 2 zachowaniami. Zdefiniuj tekstową reprezentację obiektu. Następnie utwórz 2 obiekty reprezentujące 2 telefony. Pokaż ich cechy i wywołaj ich zachowania.
2. Książki wydawane są zarówno w formie tradycyjnej (papierowej), jak i elektronicznej (e-booki). Utwórz klasę opisującą książkę, bez względu na jej typ. Następnie utwórz klasy pochodne do opisu artykułu i e-booka. Książka papierowa powinna zawierać numer jej stron, książka elektroniczna powinna zawierać nazwę pliku, w którym jest zapisana. Utwórz jedną tradycyjną książkę i jedną elektroniczną książkę. Wyświetl dane opisujące te książki.
3. Zmienna utworzona poza metodą \_\_init\_\_ jest zmienną klasy. W przeciwieństwie do zmiennej instancji, zmienna klasy przechowuje wartość, która jest wspólna (a zatem identyczna) dla wszystkich obiektów utworzonych z tej klasy.

Uruchom poniższy program. Zwróć uwagę, jak zmienna class jest deklarowana i jak jest modyfikowana. Określ zmienną klasy w programie, miejsce jej modyfikacji oraz miejsce, w którym użyto jej wartości.

# definicja  
klasy class Film(): # zmienne klasy cinema = "Multikino" def \_\_init\_\_(self, title): self.title = title def \_\_str\_\_(self): return f"{self.title} ({Film.cinema}) " # program film1 = Film("Skazani na Shawshank") print(film1) film2 = Film("Pulp Fiction") print(film2) # zmiana nazwy kina (zmiana wartości # zmiennej klasy) Film.cinema = "Kino Miasto " drukuj (film1) drukuj (film2)  
     
     
     
     
     
         
     
     
         
   

1. Student posiada imię, nazwisko, legitymację (numer albumu) oraz kierunek studiów. Wszyscy studenci studiują na tej samej uczelni (UEK Kraków). Utwórz klasę opisującą ucznia. Identyfikator ucznia powinien być przypisywany automatycznie jako sekwencyjna liczba naturalna zaczynająca się od 100000. W tym celu utwórz zmienną klasową do przechowywania numeru legitymacji ostatniego ucznia. Tworząc nowego ucznia (obiekt), zwiększ wartość tej zmiennej o jeden, a następnie użyj jej jako identyfikatora tworzonego ucznia. Następnie napisz program, który tworzy 3 różnych uczniów i wyświetla ich dane osobowe w formacie jak poniżej. Użyj metody \_\_str\_\_.

Anna MAJ (100001), Informatyka Stosowana, UEK Kraków

1. Metoda statyczna to rodzaj metody wywoływanej w kontekście klasy, a nie obiektów utworzonych z tej klasy. Poniższy program definiuje metodę statyczną.

class Arrays(): @staticmethod def print\_in\_col(array): for c in array: print(c) moja\_tablica = [4,1,8,7,2] Arrays.print\_in\_col(moja\_tablica)  
  
     
     
         
             
           

Następnie uzupełnij klasę inną statyczną metodą, która wyświetla zawartość tablicy w wierszu, oddzielając wartości przecinkami. Upewnij się, że przecinek nie jest wyświetlany po ostatniej wartości. Wywołaj nową metodę w programie.

1. W klasie Arrays dodaj trzy statyczne metody, które:
   1. utwórz tablicę z podaną liczbą elementów o tych samych wartościach. Użyj list.append():  
        
      nazwa\_metody(liczba\_elementów\_tablicy, wartość\_elementów\_tablicy)
   2. utwórz tablicę z podaną liczbą elementów i losową wartością tych elementów z zakresu <m, n>: nazwa\_metody(liczba\_elementów\_tablicy, wartość\_od, wartość\_do)
   3. określa liczbę elementów tablicy, których wartości znajdują się w podanym zakresie <m, n>:  
        
      nazwa\_metody(tablica, wartość\_z, wartość\_do)

Następnie napisz program, który utworzy 10-elementową tablicę z wartościami elementów równymi 4 oraz 20-elementową tablicę losowych liczb całkowitych z zakresu <-7,8>. Wyświetl zawartość tablic i oblicz, ile wartości z przedziału <-1,1> zawiera 20-elementowa tablica.

1. Utwórz klasę z trzema statycznymi metodami obliczania pola powierzchni figur: trójkąt, prostokąt, okrąg. Następnie użyj tych metod do obliczenia powierzchni następujących figur:
   1. Okrąg o promieniu 3
   2. Prostokąt z bokiem 4 i 7
   3. Trójkąt o podstawie 6 i wysokości 2
2. Przeciążanie operatorów to definiowanie znaczenia istniejących operatorów dla własnych typów danych. Często realizowane w formie zastosowania specjalnych metod. Na przykład w Pythonie nową funkcjonalność dla operatora porównania == można zaimplementować poprzez zdefiniowanie funkcji \_\_eq\_\_ w klasie.

W Internecie znajdź przykłady definicji \_\_eq\_\_. Następnie uzupełnij poniższą klasę Point, opisując punkt na płaszczyźnie o współrzędnych (x, y), dodając metodę \_\_eq\_\_ w celu porównania dwóch punktów.

class Point(): def \_\_init\_\_(self,x,y): self.x = x self.y = y def \_\_str\_\_(self): return f'P({self.x},{self.y})'  
     
         
         
   

Korzystając z klasy Point, utwórz program, który obliczy odległość na płaszczyźnie pomiędzy dwoma zdefiniowanymi punktami. Za pomocą instrukcji warunkowej sprawdź, czy te punkty są identyczne - użyj operatora porównania ==, czyli p1 == p2. Jeśli punkty są identyczne, wyświetl komunikat, że odległość między nimi wynosi 0. W przeciwnym razie oblicz i wyświetl odległość między dwoma punktami.

1