Inheritance

# Before Class

1. Zapoznaj się z materiałem dotyczącym zasad i technik stosowanych w programowaniu obiektowym w języku Python, zawartym w lekcjach od 54 do 59 dostępnych na kanale „Introduction to Computer Science and Programming Using Python (MIT)”.
2. Zapoznaj się z materiałami dostępnymi w witrynie w3schools dotyczącymi dziedziczenia oraz tworzenia klas pochodnych

<https://www.w3schools.com/python/python_inheritance.asp>

1. Zapoznaj się z zasadami dziedziczenia w języku Python dostępnymi w Tutorialu Pythona, punkt 9.5:

<https://docs.python.org/3/tutorial/index.html>

# During Class

## String representation of object

1. Dla wygody i czytelności kodu programu możliwe jest utworzenie tekstowej reprezentacji obiektu w postaci ciągu znaków. Taki obiekt może być następnie użyty podczas wywołania funkcji print().

Uruchom poniższy program i dokonaj jego analizy. Zwróć uwagę na zastosowaną metodę \_\_str\_\_ oraz na wywołanie funkcji print()

class University():  
   
 def \_\_init\_\_(self, name):  
 self.name = name   
   
 def \_\_str\_\_(self):  
 return self.name  
   
my\_university = University('UEK Kraków')  
print(my\_university)

1. Utwórz klasę reprezentującą utwory muzyczne. Zdefiniuj konstruktor klasy, który umożliwi ustalenie wartości początkowych utworu muzycznego (wykonawca, tytuł utworu, album, rok) w momencie tworzenia obiektu. Uzupełnij klasę o metodę \_\_str\_\_ zwracającą dane o utworze jako łańcuch znakowy, w formacie, jak poniżej.

Wykonawca: Dawid Podsiadło  
Utwór: Nie ma fal  
Album: Małomiasteczkowy  
Rok: 2018

Następnie utwórz trzy obiekty reprezentujące trzy różne utwory muzyczne. Wyświetl te obiekty.

## Class variables

1. Zmienna utworzona poza metodą \_\_init\_\_ stanowi zmienną klasy (class variable). W przeciwieństwie do zmiennej obiektu (instance variable), zmienna klasowa przechowuje wartość wspólną (a przez to identyczną) dla wszystkich obiektów utworzonych na podstawie tej klasy.

Uruchom poniższy program i dokonaj jego analizy. Zwróć uwagę na sposób deklaracji zmiennej klasowej raz sposób jej modyfukacji.

class Film():  
   
 cinema = "Multikino, Kraków"  
   
 def \_\_init\_\_(self, title):  
 self.title = title  
   
 def \_\_str\_\_(self):  
 return f"{self.title} ({Film.cinema})"  
   
film1 = Film("The Shawshank Redemption")  
print(film1)  
film2 = Film("Pulp Fiction")  
print(film2)  
  
# zmiana nazwy kina (zmiana wartości zmiennej klasowej)  
Film.cinema = "Kino Kijów, Kraków"  
print(film1)  
print(film2)

1. Student posiada imię, nazwisko, nr album oraz kierunek studiów. Wszyscy studenci studiują na tej samej uczelni (UEK Kraków). Utwórz klasę opisującą studenta, która zawierać będzie wymienione cechy. Numer albumu powinien być nadawany automatycznie, jako kolejna liczba naturalna począwszy od 100000. W tym celu utwórz zmienną klasową, w której przechowuj ostatnio nadany numer albumu. Tworząc nowego studenta, zwiększ wartość tej zmiennej, a następnie użyj jej jako numer albumu kolejnego studenta. Następnie napisz program, który utworzy 3 różnych studentów i wyświetli ich dane personalne w formacie, jak poniżej. Wykorzystaj metodę \_\_str\_\_

Wacław POTOCKI (100001), Informatyka stosowana, UEK Kraków

## Inheritnace

1. W programowaniu zorientowanym obiektowo, dziedziczenie to mechanizm współdzielenia funkcjonalności pomiędzy klasami. Klasa pochodna (dziedzicząca), utworzona na podstawie klasy bazowej, posiada swoje własne cechy i zachowania oraz dodatkowo uzyskuje (dziedziczy) cechy i zachowania również z klasy bazowej.

W grupach 2-3 osobowych uruchom poniższy program i dokonaj jego analizy. Odpowiedz na pytania:

* 1. W jaki sposób następuje określenie dziedziczenia
  2. Która klasa jest klasą bazową, a która klasą pochodną
  3. Ile cech i zachowań posiada klasa bazowa
  4. Ile własnych cech i zachowań posiada klasa pochodna
  5. Które cechy i zachowania klasa pochodna dziedziczy z klasy bazowej
  6. W jaki sposób wywołać konstruktor klasy bazowej, aby zainicjować pola w klasie bazowej
  7. W jaki sposób odwołać się do pól klasy bazowej z klasy pochodnej
  8. W jaki sposób odwołać się do metod klasy bazowej z klasy pochodnej

class Person():  
 def \_\_init\_\_(self,name):  
 self.name = name  
 def greet(self):  
 print(f'Hello everyone! I\'m {self.name}')  
  
class Teacher(Person):  
 def \_\_init\_\_(self,name,university):  
 self.university = university  
 super().\_\_init\_\_(name)  
 def say(self):  
 print(f'I work as a teacher at {self.university}')  
 def bye(self):  
 print(f'And now {self.name} is telling you goodbye!')  
   
t = Teacher('Johnny','UEK')  
t.greet()  
t.say()  
t.bye()

1. Aaaxcfb sdgfb dgb dgb

## Operator overloading

1. Przeciążanie operatorów to definiowanie operatorów dla własnych typów danych. Często realizowane w postaci użycia specjalnych metod. Przykładowo, w języku Python nową funkcjonalność dla operatora porównania == można zrealizować definiując w klasie funkcję \_\_eq\_\_.

W grupach 2-3 osobowych, odszukaj w Internecie przykłady definicji funkcji \_\_eq\_\_. Następnie uzupełnij poniższą klasę Point, opisującą punkt na płaszczyźnie o współrzędnych x,y, dodając metodę \_\_eq\_\_ umożliwiającą porównanie 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})'

Wykorzystując klasę Point napisz program, który dla dwóch zdefiniowanych punktów obliczy ich odległość na płaszczyźnie. Korzystając z instrukcji warunkowej sprawdź, czy punkty te są identyczne – użyj operatora porównania ==, tj. p1==p2. Jeśli punkty są identyczne, wyświetl komunikat, że odległość pomiędzy nimi wynosi 0. W przeciwnym wypadku oblicz i wyświetl odległość pomiędzy tymi dwoma punktami.

## Static methods

1. Metoda statyczna to rodzaj metody, która jest wywoływana w kontekście klasy, a nie obiektów tworzonych na podstawie tej klasy. Poniższy program zawiera definicję metody statycznej. W grupach 2-3 osobowych zwróć uwagę na sposób jej deklaracji. Jakie dostrzegasz różnice w stosunku do metody instancyjnej? Wskaż je w poniższym programie.

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

Następnie uzupełnij klasę o kolejną metodę statyczną, która wyświetla zawartość tablicy w wierszu, rozdzielając wartości znakiem przecinka. Zwróć uwagę, aby po ostatniej wartości przecinek nie był wyświetlany.

1. Uzupełnij klasę arrays o metody statyczne, które realizować będą:
   1. Tworzenie tablicy o zadanej liczbie elementów oraz zadanej, jednakowej wartości tych elementów. Wykorzystaj list.append()  
      metoda(liczba\_elementow\_tablicy,wartosc\_elementow\_tablicy)
   2. Tworzenie tablicy o zadanej liczbie elementów oraz losowej wartości tych elementów ze wskazanego przedziału <m,n>  
      metoda(liczba\_elementow\_tablicy,wartosc\_od,wartosc\_do)
   3. Wyznaczenie liczby elementów tablicy, których wartości znajdują się w zadanym przedziale <m,n>  
      metoda(tablica,wartość\_od,wartość\_do)

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

# After Class

1. Utwórz klasę opisującą telefony komórkowe zawierającą co najmniej 3 cechy oraz 2 zachowania telefonów. Zdefiniuj tekstową reprezentację obiektu. Następnie utwórz 2 obiekty reprezentujące 2 telefony. Wyświetl ich charakterystykę (cechy) oraz wykonaj zdefiniowane metody.
2. Utwórz program do ewidencji figur geometrycznych. W tym celu utwórz klasę opisującą ogólną figurę geometryczną. Klasa ta powinna posiadać jedną cechę opisującą nazwę figury. Następnie utwórz klasy pochodne opisujące koło, kwadrat oraz trójkąt. Klasy te powinny zawierać zmienne charakteryzujące te figury oraz umożliwiające obliczenie pola jej powierzchni (np. w przypadku trójkąta: a, b, h). Nazwę figury oraz wartości liczbowe należy przekazać w momencie tworzenia obiektu reprezentującego figurę. Każda klasa powinna posiadać również metodę obliczającą pole powierzchni figur oraz reprezentację tekstową obiektu, np.

Koło o promieniu 4

Utwórz w programie po dwie figury każdego rodzaju i obiekty te umieść w tablicy.

1. Napisz program realizujący funkcjonalność wypożyczalni pojazdów.

Zdefiniuj klasę opisującą pojedynczy pojazd, który powinien posiadać cechy: marka pojazdu, numer rejestracyjny, przebieg w km oraz czy pojazd jest wypożyczony. Marka pojazdu oraz numer rejestracyjny, powinna zostać zdefiniowana w momencie jego tworzenia. Pojazd w momencie tworzenia jest domyślnie niewypożyczony i posiada przebieg początkowy równo 0. Dodaj możliwość wyświetlenia wymienionych danych pojazdu. Wykorzystaj tekstową reprezentację obiektu. Dodaj także metody umożliwiające zmianę przebiegu pojazdu oraz statusu pojazdu (czy jest wypożyczony, czy dostępny).

Zdefiniuj klasę opisującą wypożyczalnię. Powinna ona zawierać nazwę oraz listę posiadanych pojazdów (użyj tablicy). Dodaj metodę umożliwiającą dodanie pojazdu do wypożyczalni. Dodaj również metodę wyświetlającą nazwę wypożyczalni oraz numerowaną wszystkich listę pojazdów w wypożyczalni. Wykorzystaj tekstową reprezentację obiektu. Dodaj także metodę wyświetlającą numerowaną listę pojazdów w wypożyczalni, które obecnie są dostępne dla klientów (nie są wypożyczone) oraz metodę wyświetlającą listę pojazdów wypożyczonych. Dodaj również metodę umożliwiającą zmianę statusu wypożyczenia pojazdu o podanym numerze rejestracyjnym. W przypadku zwrotu pożyczonego pojazdu (gdy samochód zmienia status z wypożyczonego na niewypożyczony), możliwe jest wprowadzenie liczby km przejechanych podczas wypożyczenia, która to wartość powinna zwiększyć całkowity przebieg pojazdu.

Na podstawie klasy opisującej pojazd, utwórz klasy pochodne opisujące samochód osobowy oraz samochód dostawczy. Samochód osobowy posiada cechę liczba miejsc, natomiast samochód dostawczy charakteryzuje się ładownością w tonach. Dodaj możliwość wyświetlenia wszystkich danych zarówno dla pojazdu osobowego oraz dostawczego. Wykorzystaj tekstową reprezentację obiektu.

Wykorzystując zdefiniowane klasy, napisz program, w którym:

* 1. Utwórz wypożyczalnię pojazdów
  2. Dodaj do wypożyczalni 3 pojazdy osobowe oraz 2 pojazdy dostawcze
  3. Wyświetli listę wszystkich pojazdów w wypożyczalni
  4. Wypożycz jeden pojazd osobowy oraz jeden pojazd dostawczy.
  5. Wyświetl listę wypożyczonych pojazdów.
  6. Wyświetl listę pojazdów do wypożyczenia.
  7. Zwróć pojazd osobowy podając 950 przejechanych km podczas wypożyczenia
  8. Wyświetl listę wypożyczonych pojazdów.
  9. Wyświetl listę pojazdów do wypożyczenia.