OBJECT ORIENTED PROGRAMMING

BEFORE CLASS

1. Zapoznaj się z materiałem dotyczącym zasad i technik stosowanych w programowaniu zorientowanym obiektowo w języku Python - lekcje od 54 do 59 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

3. 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

4. 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().

   W grupach 2-3 osobowych 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)
```

5. 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 (4 wiersze).

```
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

6. 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.

   W grupie 2-3 osobowej uruchom poniższy program i dokonaj jego analizy. Zwróć uwagę na sposób deklaracji zmiennej klasowej oraz sposób jej modyfikacji. Wskaż w programie zmienną klasową, miejsce gdzie jest modyfikowana oraz miejsce, gdzie została użyta jej wartość.

```
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)
```

7. Student posiada imię, nazwisko, nr albumu 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 (obiekt), zwiększ wartość tej zmiennej, a następnie użyj jej jako numer albumu

dla tworzonego 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
```

Inheritance

8. 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:

   a. W jaki sposób następuje określenie dziedziczenia
   b. Która klasa jest klasą bazową, a która klasą pochodną
   c. Ile cech i zachowań posiada klasa bazowa
   d. Ile własnych cech i zachowań posiada klasa pochodna
   e. Które cechy i zachowania klasa pochodna dziedziczy z klasy bazowej
   f. W jaki sposób wywołać konstruktor klasy bazowej, aby zainicjować pola w klasie bazowej
   g. W jaki sposób odwołać się do pól klasy bazowej z klasy pochodnej
   h. 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()
```

9. Poniższa klasa opisuje dowolny komunikat tekstowy.

```
class Message():
    def __init__(self):
```

```
        self.message = ''
    def set_message(self,message):
```

Uzupełnij definicję metody set_message(), aby w momencie ustawiania treści komunikatu jego pierwsza litera została zamieniona na wielką, a pozostałe na małe. Ponadto na końcu komunikatu dodaj zwrot pożegnalny BYE.

Następnie utwórz dwie klasy pochodne: SMS oraz Email. Klasa SMS powinna zawierać właściwość opisującą nr telefonu, natomiast klasa Email powinna zawierać właściwości: adres nadawcy, adres odbiorcy oraz temat listu. W obydwu klasach zdefiniuj metodę send(), która służy do wysyłania wiadomości. Wysłanie wiadomości polega na wyświetleniu danych charakterystycznych dla danego typu wiadomości oraz jej treści, np.:

```
Wysyłanie emaila...
Od:    nowak@onet.pl
Do:    kowalski@gmail.com
Temat: Spotkanie w czwartek
Treść: Informuję, że nasze czwartkowe spotkanie zostało
odwołane. BYE.
```

Utwórz i wyślij jeden email oraz jeden SMS.

Operator overloading

10. Przeciążanie operatorów to definiowanie znaczenia istniejących 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__ w języku Python. 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, iż odległość pomiędzy nimi wynosi 0. W przeciwnym wypadku oblicz i wyświetl odległość pomiędzy tymi dwoma punktami.

Static methods

11. 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. Wywołaj nową metodę dla podanej tablicy.

12. Uzupełnij klasę arrays o metody statyczne, które realizować będą:
    a. Tworzenie tablicy o zadanej liczbie elementów oraz zadanej, jednakowej wartości tych elementów. Wykorzystaj list.append()
       `metoda(liczba_elementow_tablicy,wartosc_elementow_tablicy)`
    b. Tworzenie tablicy o zadanej liczbie elementów oraz losowej wartości tych elementów ze wskazanego przedziału <m,n>
       `metoda(liczba_elementow_tablicy,wartosc_od,wartosc_do)`
    c. 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.

13. Uzupełnij klasę arrays o zmienną statyczną zawierającą znak separatora do oddzielania elementów tablicy. Ustal przecinek jako wartość początkową. Dodaj metodę statyczną, umożliwiającą zmianę znaku separatora. Następnie zmodyfikuj działanie metod wyświetlających zawartość tablicy, aby rozdzielały wyświetlane wartości znakiem separatora zawartym w zmiennej statycznej.

AFTER CLASS

14. 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.

15. Książki wydawane są zarówno w postaci tradycyjnej (papierowej), jak i elektronicznej (ebooki). Utwórz klasę opisującą książkę, bez względu na jej rodzaj. Następnie, utwórz klasy pochodne dla opisu książki papierowej oraz książki elektronicznej. Książka papierowa powinna zawierać liczbę jej stron, natomiast książka elektroniczna nazwę pliku, w którym jej ona zapisana. Utwórz jedną książkę tradycyjną oraz jedną elektroniczną. Wyświetl dane tych książek.

16. Funkcja sorted() oraz metoda sort() pozwalają na porządkowanie wartości https://docs.python.org/3/howto/sorting.html

    Utwórz klasę opisującą studenta (imię, nazwisko, nr albumu). Dodaj tekstową reprezentację obiektu tej klasy. Jednocześnie dodaj w klasie metody __eq__ oraz __lt__ przeciążające operatory == oraz <= umożliwiające porównanie obiektów pod kątem numeru albumu studenta.

    Korzystając ze zdefiniowanej klasy, utwórz poniższą listę studentów w podanej kolejności. Wyświetl utworzoną listę studentów. Następnie dokonaj jej sortowania przy użyciu funkcji sorted(). Wyświetl posortowaną listę studentów wg numeru albumu.

```
Anna Tomaszewska 141820
Wojciech Zbych 201003
Maja Kowalska 153202
Marek Migiel 138600
```

17. Utwórz klasę zawierającą trzy metody statyczne dla obliczania pola powierzchni: trójkąta, prostokąta, koła. Użyj tych metod do obliczenia pola powierzchni:

    a. Koła o promieniu 3
    b. Prostokąta o bokach 4 oraz 7
    c. Trójkąta o podstawie 6 i wysokości 2

18. 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:

a. Utwórz wypożyczalnię pojazdów
b. Dodaj do wypożyczalni 3 pojazdy osobowe oraz 2 pojazdy dostawcze
c. Wyświetli listę wszystkich pojazdów w wypożyczalni
d. Wypożycz jeden pojazd osobowy oraz jeden pojazd dostawczy.
e. Wyświetl listę wypożyczonych pojazdów.
f. Wyświetl listę pojazdów do wypożyczenia.
g. Zwróć pojazd osobowy podając 950 przejechanych km podczas wypożyczenia
h. Wyświetl listę wypożyczonych pojazdów.
i. Wyświetl listę pojazdów do wypożyczenia.