

# CLASSES AND OBJECTS

---

## BEFORE CLASS

1. Zapoznaj się z materiałem omawiającym czym jest język programowania zorientowany obiektowo

<https://www.youtube.com/watch?v=SS-9y0H3Si8&list=PLck2iFB-00xnYImq1V4CSN9I4JdSCiojg&index=7&t=0s>

2. Zapoznaj się z materiałem wprowadzającym do programowania zorientowanego obiektowo

<https://www.youtube.com/watch?v=0yEBBCfalZk>

3. Zapoznaj się z materiałem wprowadzającym do programowania zorientowanego obiektowo w języku Python, zawartym w lekcjach od 50 do 53 dostępnych na kanale „Introduction to Computer Science and Programming Using Python (MIT)”.

4. Zapoznaj się z materiałami dostępnymi w witrynie w3schools dotyczącymi tworzenia klas i ich składowych (pól i metod)

[https://www.w3schools.com/python/python\\_classes.asp](https://www.w3schools.com/python/python_classes.asp)

5. Zapoznaj się z definicją klas, pól i metod, dostępną w Tutorialu Pythona, punkt 9.3:

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

## DURING CLASS

6. W grupach 2-3 osobowych wymień co najmniej 3 cechy oraz 3 zachowania dla każdej z poniższych klas obiektów.

- a. Książka
- b. Połączenie telefoniczne
- c. Grupa studentów

Zapisz te nazwy w dokumencie tekstowym.

Dla wymienionych cech i zachowań utwórz odpowiadające im nazwy pól i metod. Zastosuj konwencję nazw „snake case” ([https://simple.wikipedia.org/wiki/Snake\\_case](https://simple.wikipedia.org/wiki/Snake_case)). Pamiętaj o stosowaniu czasowników w nazwach metod.

7. Napisz program, w którym utwórz klasę University, składającą się z jednego pola zawierającego nazwę uczelni oraz jednej metody wyświetlającej tę nazwę.

```
class University():  
  
    # konstruktor obiektu (metoda __init__)  
    def __init__(self):  
        # cechy obiektu (pola)  
        self.name = 'UEK'  
  
    # zachowania obiektu (metody)  
    def print_name(self):  
        print(self.name)
```

Następnie utwórz obiekt klasy University i wywołaj metodę, aby wyświetlić nazwę uczelni.

8. Dodaj w klasie University metodę set\_name(), która umożliwi zmianę nazwy uczelni (zmianę wartości pola).

```
def set_name(self, new_name):  
    self.name = new_name
```

Następnie zmodyfikuj program, aby zmienić nazwę uczelni w utworzonym obiekcie (wprowadź AGH) oraz wyświetlić tę zmienioną nazwę.

9. Dodaj w klasie University pole fullname zawierające pełną nazwę uczelni oraz metody print\_fullname() i set\_fullname() umożliwiające wyświetlanie i zmianę pełnej nazwy uczelni. Następnie utwórz obiekt i wyświetl początkową pełną nazwę uczelni oraz zmień i wyświetl jej nową pełną nazwę.

10. Napisz program, w którym utwórz klasę TV opisującą odbiorniki telewizyjne (telewizory). Klasa powinna zawierać jedno pole typu logicznego o nazwie 'is\_on' określające, czy telewizor jest załączony. Domyślnie telewizor jest wyłączony. Dodaj w klasie metody on() oraz off() umożliwiające odpowiednio załączanie oraz wyłączanie telewizora. Dodaj również metodę show\_status() wyświetlającą informację, czy telewizor jest załączony, czy wyłączony. Przykładowy komunikat:

✓ Telewizor jest załączony

Następnie w programie sprawdź działanie telewizora:

- ✓ a. Utwórz telewizor
- ✓ b. Wyświetl stan telewizora
- ✓ c. Załącz telewizor
- ✓ d. Wyświetl stan telewizora
- ✓ e. Wyłącz telewizor
- ✓ f. Wyświetl stan telewizora

11. Rozszerz klasę TV dodając pole channel\_no wskazujące na nr kanału telewizyjnego wyświetlanego przez odbiornik. Początkowo odbiornik TV ustawiony jest na kanał 1.

Zmodyfikuj metodę `show_status()`, aby wyświetlała ona również nr kanału telewizora, ale tylko w przypadku, gdy telewizor jest załączony, np.

Odbiornik TV jest załączony, kanał 1 ✓

Następnie sprawdź działanie telewizora.

12. Dodaj w klasie TV metodę `set_channel(new_channel_no)` umożliwiającą ustawienie numeru kanału telewizyjnego. Następnie sprawdź działanie telewizora:

- ✓ a. Utwórz telewizor
- ✓ b. Wyświetl stan telewizora
- ✓ c. Załącz telewizor
- ✓ d. Wyświetl stan telewizora
- ✓ e. Zmień kanał na piąty
- ✓ f. Wyświetl stan telewizora
- ✓ g. Wyłącz telewizor
- ✓ h. Wyświetl stan telewizora

13. Dodaj w klasie TV pole `channels` zawierające listę dostępnych nazw kanałów telewizyjnych (tablica). Początkowo tablica powinna być pusta (telewizor niezaprogramowany, brak dostępnych kanałów). Dodaj w klasie TV metody `set_channels(channels_list)` oraz `show_channels()`, które umożliwią zaprogramowanie kanałów w telewizorze oraz wyświetlenie listy dostępnych kanałów. Przykładowa lista dostępnych kanałów:

LISTA KANAŁÓW TV

- 1. TVP1
- 2. TVP2
- 3. Polsat
- 4. TVN
- 5. Filmbox

Następnie sprawdź działanie telewizora:

- a. Utwórz telewizor
- b. Wyświetl stan telewizora
- c. Załącz telewizor
- d. Wyświetl stan telewizora
- e. Wyświetl listę dostępnych kanałów
- f. Zaprogramuj kanały: TVP1, TVP2, Polsat, TVN, Filmbox
- g. Wyświetl listę dostępnych kanałów
- h. Wyświetl status telewizora
- i. Wyłącz telewizor

14. W klasie TV wprowadź zmiany w metodzie `show_status()` aby wyświetlała ona nie tylko nr wybranego kanału, ale również jego nazwę. Przykładowo: jeśli telewizor posiada zaprogramowane kanały: TVP1, TVP2, Polsat, TVN, Filmbox, to jeśli wybrany nr kanału to 4, powinna wyświetlić się nazwa TVN. Gdy nr wybranego kanału wykracza poza listę dostępnych kanałów, nazwa kanału nie jest wyświetlana.

Odbiornik TV jest załączony, kanał 4 (TVN)

Następnie sprawdź działanie telewizora. Zaprogramuj co najmniej 7 kanałów (siedem stacji telewizyjnych) oraz zmieniaj numery kanałów i każdorazowo wyświetlaj status telewizora.

15. W klasie TV dodaj obsługę regulacji głośności w zakresie o 0 do 10. Początkowa wartość poziomu głośności wynosi 0. Dodaj dwie metody umożliwiające zwiększanie i zmniejszanie poziomu głośności telewizora o jeden stopień. Zwróć uwagę, iż nie można zwiększyć czy zmniejszyć głośności poza podany zakres. Bieżący poziom głośności wyświetlaj w metodzie `show_status()`. Następnie sprawdź działanie telewizora regulując i wyświetlając jego poziom głośności.
16. Książka elektroniczna (e-book) to książka w formie cyfrowej, która może być czytana przy użyciu komputera lub innych urządzeń elektronicznych (czytników książek elektronicznych). Napisz program, w którym zdefiniuj klasę opisującą cechy i zachowania książki elektronicznej. Każda książka posiada tytuł, autora, liczbę stron oraz nr bieżącej strony, która aktualnie jest czytana. Możliwe jest otwarcie książki – wtedy możemy ją czytać. Jeśli książka jest otwarta, możliwe jest przejście do kolejnej lub poprzedniej strony.

Klasę opisującą książki umieść w odrębnym pliku, który zaimportuj do głównego programu. W głównym pliku programu wykonaj operację na książce:

- a. Utwórz książkę – nadaj jej tytuł, autora, liczbę stron (sprawdź, w jaki sposób ustawić wartości początkowe pól w momencie tworzenia obiektu wykorzystując metodę/konstruktor `__init__` i przekazując wartości początkowe jako argumenty wywołania metody)
- b. Otwórz książkę
- c. Wyświetl status książki (tytuł, autor, liczba stron, nr bieżącej strony)
- d. Przeczytaj kilka stron
- e. Wyświetl status książki
- f. Zamknij książkę
- g. Przeczytaj kilka stron (nie powinno być możliwości wykonania tej operacji – wyświetl komunikat, że książka jest zamknięta).

## AFTER CLASS

17. Napisz program zawierający klasę definiującą ułamek zwykły. Zdefiniuj klasę opisującą ułamek zwykły, zawierającą pola licznik oraz mianownik, a także metody służące do tworzenia ułamka, jego upraszczania oraz wyświetlania). Sprawdź działanie klasy tworząc poniższe ułamki zwykłe oraz wyświetlając je na konsoli.

1/2  
12/21

18. Utwórz klasę opisującą kostkę do gry. Klasa powinna zawierać metodę (w nazwie zastosuj czasownik) zwracającą wyrzuconą przez kostkę liczbę oczek, tj. wartość naturalną  $<1,6>$ . Następnie napisz program, w którym utwórz trzy kostki. Rzuć tymi trzema kostkami wyświetlając liczbę wyrzuconych oczek na każdej z kostek oraz sumę wyrzuconych oczek. Powtórz operację kilkakrotnie sprawdzając, czy obliczona suma zgodna jest z liczbą wyrzuconych oczek. Zastosuj instrukcję iteracyjną.

19. Rachunek bankowy posiada 26 cyfrowy numer nadawany podczas tworzenia rachunku. Początkowe saldo rachunku wynosi 0 zł. Na rachunek można wpłacać dowolną kwotę. Z rachunku można też wypłacać dowolną kwotę pod warunkiem, iż nie jest ona większa od salda rachunku. Próba wypłacenia większej kwoty powoduje wyświetlenie komunikatu: „Niewystarczająca ilość środków na rachunku”. W każdym momencie możliwe jest wyświetlenie informacji o numerze i stanie rachunku bankowego w formacie:

```
Rachunek nr: 11 1111 1111 1111 1111 1111 1111
Saldo: 25,38 zł
```

Utwórz program realizujący obsługę rachunku bankowego.

- Zapoznaj się uważnie z opisywanym problemem.
- Zidentyfikuj obiekt.
- Określ właściwości oraz zachowania obiektu.
- Dokonaj transformacji właściwości i zachowań obiektu do pól i metod klasy, która stanowić będzie wzorzec dla tworzenia obiektu.
- Utwórz szkielet klasy bez tworzenia zawartości metod.
- Utwórz zawartość poszczególnych metod.

Sprawdź działanie programu:

- Utwórz rachunek bankowy o numerze 12 3456 5555 9090 1111 0000 7722
  - Wyświetl informację o saldzie rachunku.
  - Wpłać 25,30 zł
  - Wyświetl informację o saldzie rachunku.
  - Wypłać 31,70 zł
  - Wyświetl informację o saldzie rachunku.
  - Wypłać 14 zł
  - Wyświetl informację o saldzie rachunku.
20. Usługa <https://www.flightradar24.com/> dostarcza informacje w czasie rzeczywistym o ruchu w przestrzeni powietrznej. Wykorzystując ten serwis zobacz, jakie samoloty przelatują obecnie w okolicach Krakowa. Spróbuj odczytać dane kilku samolotów (rodzaj statku powietrznego, numer lotu, prędkość, wysokość lotu, lotnisko docelowe). Następnie utwórz model samolotu w programie komputerowym stosując podejście obiektowe.

Każdy samolot odbywający lot posiada przypisany numer lotu, np. BA226. W momencie startu samolot zwiększa swą wysokość od 0 do podanej w momencie startu wysokości z przedziału 1000 do 2000m. W trakcie lotu samolot może zmienić wysokość lotu (zwiększyć lub zmniejszyć) jednakże tylko w zakresie od 300m do 11000m. Próba zmiany wysokości lotu poza podanym zakresem powoduje wyświetlenie komunikatu ostrzegawczego. Samolot może wykonać operację lądowania. Samolot może rozpocząć lądowanie, gdy jego wysokość lotu jest mniejsza od 500m. Gdy wysokość lotu jest wyższa, wyświetlany jest komunikat: „Zbyt duża wysokość dla lądowania. Obniż lot”. W każdym momencie można poprosić pilota o podanie podstawowych informacji o locie. Wyświetlany jest wtedy komunikat, zgodny z formatem:

```
Tu BA226, moja wysokość lotu to 7500m
```

Utwórz program sterujący lotem samolotu.

- a. Zapoznaj się uważnie z opisywanym problemem.
- b. Zidentyfikuj obiekt.
- c. Określ właściwości oraz zachowania obiektu.
- d. Dokonaj transformacji właściwości i zachowań obiektu do pól i metod klasy, która stanowić będzie wzorzec dla tworzenia obiektu. Utwórz szkielet klasy bez tworzenia zawartości metod.
- e. Utwórz zawartość poszczególnych metod.

Następnie sprawdź działanie utworzonej klasy:

- f. Utwórz samolot, który realizował będzie lot o numerze LOT881
- g. Wystartuj samolot.
- h. Poproś pilota o podanie informacji o locie.
- i. Zwiększ wysokość lotu do 8900m
- j. Poproś pilota o podanie informacji o locie.
- k. Zmniejsz wysokość lotu do 200m.
- l. Poproś pilota o podanie informacji o locie.
- m. Wyląduj samolotem.
- n. Poproś pilota o podanie informacji o locie.

21. Napisz program zawierający klasę Statystyka opisującą właściwości dowolnego zbioru liczb. Klasa powinna umożliwiać:

- a. Dodanie do zbioru liczb kolejnej liczby odczytanej z klawiatury (liczby przechowuj w tablicy)
- b. Wyświetlenie wszystkich liczb oddzielonych znakiem odstępu
- c. Wyznaczenie liczby największej
- d. Wyznaczenie liczby najmniejszej
- e. Obliczenie średniej arytmetycznej liczb
- f. Obliczenie mediany
- g. Wyświetlenie obliczonych/wyznaczonych wielkości statystycznych (minimum, maksimum, średnia, mediana)

Sprawdź działanie programu dla liczb:

12, 37, 6, 9, 17

22. Termometr lekarski umożliwia pomiar temperatury pacjenta w zakresie od 34.0 do 42.0 stopni Celsjusza, z dokładnością do 0.1 stopnia. Napisz program, w którym zdefiniuj klasę opisującą cechy i zachowania termometru. Termometr powinien umożliwiać pomiar temperatury (zrealizuj to generując liczbę losową z przedziału 34.0 do 42.0) oraz wyświetlanie zmierzonej wartości (zapamiętaj odczytaną wartość temperatury w polu 'temperature'). Jeśli temperatura wynosi co najmniej 37 stopni Celsjusza, termometr powinien wyświetlić dodatkowo komunikat 'Gorączka', np.

Zmierzona temperatura: 37.2C (gorączka)

Gdy temperatura wynosi co najmniej 41.0, termometr powinien dodatkowo wyświetlić komunikat 'Stan zagrożenia życia!!!'. Definicję klasy oraz program główny umieść w oddzielnych plikach. Sprawdź działanie programu wykonując poniższe czynności:

- a. Utwórz termometr
  - b. Załącz termometr
  - c. Zmierz temperaturę
  - d. Wyświetl zmierzoną temperaturę
  - e. Wyłącz termometr
23. Klasa Kontakt zawiera pola 'nazwa', 'email', 'telefon' umożliwiające opisanie pojedynczego kontaktu w smartfonie. Klasa ListaKontaktow umożliwia przechowywanie kontaktów (obiekty opisujące kontakty przechowuj w tablicy) oraz wykonywanie operacji:
- a. Dodanie nowego kontaktu
  - b. Wyświetlenie listy kontaktów

Napisz program, w którym utwórz obiekt reprezentujący listę kontaktów oraz dodaj do niej dane:

Kowalski Jan	jank@onet.pl	555234000
Borek Patrycja	bp@o2.pl	232000199
Maj Piotr	maj@google.pl	222999100
Adamczyk Anna	ada@poczta.pl	100200300

Wyświetl listę kontaktów dostępną w smartfonie.