**Klasy i przedmioty**

# PRZED ZAJĘCIAMI

1. Zapoznaj się z koncepcją programowania obiektowego. Następnie obejrzyj wideo:

[https://www.youtube.com/watch?v=SS-9y0H3Si8&list=PLck2iFB-00xnYlmq1V4CSN9I4JdSCiojg&index=7&t=0s](https://translate.google.com/translate?hl=pl&prev=_t&sl=en&tl=pl&u=https://www.youtube.com/watch%3Fv%3DSS-9y0H3Si8%26list%3DPLck2iFB-00xnYlmq1V4CSN9I4JdSCiojg%26index%3D7%26t%3D0s)

1. Jak definiuje się klasy w Pythonie? Obejrzyj wideo:

[https://youtube.com/playlist?list=PLi01XoE8jYohWFPpC17Z-wWhPOSuh8Er-](https://translate.google.com/translate?hl=pl&prev=_t&sl=en&tl=pl&u=https://youtube.com/playlist%3Flist%3DPLi01XoE8jYohWFPpC17Z-wWhPOSuh8Er-)

1. Zapoznaj się z wprowadzeniem do programowania obiektowego w Pythonie zawartym w lekcjach od 50 do 53 dostępnym na kanale „Wprowadzenie do informatyki i programowania w Pythonie (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 tutorialami dostępnymi na w3schools, które dotyczą tworzenia klas i ich komponentów (pól i metod).

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

1. Zapoznaj się z definicjami klas, pól i metod w tutorialu Pythona 9.3

[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

1. Zidentyfikuj co najmniej 3 stany i 3 zachowania dla każdego z poniższych obiektów.
   1. Książka
   2. Połączenie telefoniczne
   3. Grupa studentów

Następnie dla wymienionych stanów i zachowań utwórz odpowiednie pola (atrybuty) i metody. Pamiętaj, aby w nazwach metod używać czasowników, ponieważ opisują one czynności.

1. Napisz program, w którym utworzysz klasę Uczelni, składającą się z jednego pola zawierającego nazwę Uczelni oraz jednej metody wyświetlania tej nazwy.

class University(): # konstruktor obiektu (metoda \_\_init\_\_) def \_\_init\_\_(self): # stany/atrybuty obiektów (pola) self.name = 'CUE' # zachowania obiektów (metody) def print\_name(self): print(self.name )  
  
     
     
         
             
  
     
     

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

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

def set\_name(self, name): self.name = name

Następnie zmodyfikuj program, aby zmienić nazwę uczelni w tworzonym obiekcie. Wpisz MIT, co oznacza Massachusetts Institute of Technology i wyświetl zmienioną nazwę.

1. Dodaj pole fullname do klasy University, które zawiera pełną nazwę uniwersytetu oraz metody print\_fullname() i set\_fullname(), aby wyświetlić i zmienić pełną nazwę uniwersytetu. Następnie utwórz obiekt i:
   1. wyświetl krótką nazwę uniwersytetu
   2. wyświetl pełną nazwę uniwersytetu
   3. zmień nazwę uniwersytetu
   4. zmień pełną nazwę uczelni
   5. wyświetl krótką nazwę uniwersytetu
   6. wyświetl pełną nazwę uniwersytetu
2. Napisz program, w którym tworzysz zajęcia telewizyjne opisujące telewizor. Klasa powinna zawierać jedno pole logiczne o nazwie „is\_on”, które określa, czy telewizor jest włączony. Domyślnie telewizor jest wyłączony. Dodaj metody turn\_on() i turn\_off() w klasie, aby odpowiednio włączać i wyłączać telewizor. Dodaj również metodę show\_status(), aby wyświetlić, czy telewizor jest włączony, czy wyłączony. Przykładowa wiadomość:

Telewizor jest włączony

Następnie spróbuj użyć telewizora w programie:

1. Utwórz telewizor
2. Pokaż stan telewizora
3. Włącz telewizor
4. Pokaż stan telewizora
5. Wyłącz telewizor
6. Pokaż stan telewizora
7. W klasie TV dodaj pole channel\_no wskazujące numer kanału TV wyświetlanego przez telewizor. Początkowo telewizor jest ustawiony na kanał 1. Zmodyfikuj metodę show\_status() tak, aby wyświetlała również numer kanału telewizyjnego, ale tylko wtedy, gdy telewizor jest włączony:

Telewizor jest włączony, kanał 1

Następnie spróbuj użyć telewizora .

1. Dodaj metodę set\_channel(new\_channel\_no) w klasie TV, aby ustawić numer kanału TV. Następnie spróbuj użyć telewizora.
   1. Stwórz telewizor
   2. Pokaż stan telewizora
   3. Włącz telewizor
   4. Pokaż stan telewizora
   5. Zmień kanał telewizyjny na 5
   6. Pokaż stan telewizora
   7. Wyłącz telewizor
   8. Pokaż stan telewizora
2. W klasie TV dodaj pole kanałów zawierające listę dostępnych nazw kanałów telewizyjnych (tablicę). Początkowo tablica powinna być pusta (telewizor nie zaprogramowany, brak dostępnych kanałów). Dodaj metody set\_channels(channels\_list) i show\_channels() w klasie TV, co pozwala ustawić kanały w telewizorze i wyświetlić listę dostępnych kanałów. Przykładowa lista dostępnych kanałów:

Lista kanałów:  
1. TVP1  
2. TVP2  
3. Polsat  
4. TVN  
5. Filmbox  
6. Discovery

Następnie spróbuj użyć telewizora:

1. Stwórz telewizor
2. Pokaż stan telewizora
3. Włącz telewizor
4. Pokaż stan telewizora
5. Wyświetl listę dostępnych kanałów
6. Ustaw kanały TV: TVP1, TVP2, Polsat, TVN, Filmbox, Discovery
7. Wyświetl listę dostępnych kanałów
8. Pokaż stan telewizora
9. Wyłącz telewizor
10. W klasie TV wprowadź zmiany w metodzie show\_status() tak, aby wyświetlała nie tylko wybrany numer kanału, ale także jego nazwę. Gdy wybrany numer kanału przekracza listę dostępnych kanałów, nazwa kanału nie jest wyświetlana.

TV jest włączony, kanał 4 (TVN)

Następnie spróbuj użyć telewizora. Ustaw co najmniej 7 kanałów (siedem stacji TV), zmieniaj numery kanałów i wyświetlaj status TV za każdym razem.

# PO ZAJĘCIACH

1. W klasie TV dodaj obsługę regulacji głośności w zakresie od 0 do 10. Początkowa wartość poziomu głośności to 0. Dodaj dwie metody zwiększania i zmniejszania poziomu głośności telewizora o jeden. Pamiętaj, że nie możesz zwiększać ani zmniejszać głośności poza określony zakres. Wyświetla aktualny poziom głośności w metodzie show\_status(). Następnie sprawdź działanie telewizora, dostosowując i wyświetlając jego poziom głośności.
2. E-book to książka cyfrowa, którą można czytać za pomocą komputera lub innych urządzeń elektronicznych (elektronicznych czytników książek). Napisz program, w którym zdefiniujesz klasę opisującą stany i zachowanie e-booka. Każda książka ma tytuł, autora, liczbę stron oraz numer aktualnie czytanej strony. Można otworzyć książkę - wtedy możemy ją przeczytać. Jeśli książka jest otwarta, można przejść do następnej lub poprzedniej strony.

Umieść klasę opisującą e-booki w osobnym pliku/module. W głównym pliku programu spróbuj użyć e-booka:

1. Stwórz książkę z tytułem, autorem, ilością stron (sprawdź jak ustawić początkowe wartości pól w momencie tworzenia obiektu za pomocą metody / konstruktora \_\_init\_\_ i przekazując początkowe wartości jako argumenty do wywołania metody)
2. Otwórz książkę
3. Wyświetl status książki (tytuł, autor, numery stron, numer bieżącej strony)
4. Przeczytaj kilka stron
5. Wyświetl stan książki
6. Zamknij książkę
7. Przeczytaj kilka stron (nie powinno być możliwe wykonanie tej operacji - wyświetl komunikat, że książka jest zamknięta).
8. Termometr medyczny mierzy temperaturę pacjenta w zakresie od 34,0 do 42,0 stopni Celsjusza, z dokładnością do 0,1 stopnia. Napisz program, w którym zdefiniujemy klasę opisującą stany i zachowanie termometru. Termometr powinien umożliwiać pomiar temperatury (zrób to generując losową liczbę z zakresu od 34,0 do 42,0) i wyświetlaj zmierzoną wartość. Jeśli temperatura wynosi co najmniej 37 stopni Celsjusza, termometr powinien dodatkowo wyświetlić komunikat „Gorączka” np.

Temperatura: 37,2C (gorączka)

Gdy temperatura wynosi co najmniej 41,0 termometr powinien dodatkowo wyświetlić komunikat „TEMPERATURA KRYTYCZNA!!”. Umieść definicję klasy i główny program w osobnych plikach. Następnie użyj programu i:

1. Utwórz termometr
2. Włącz termometr
3. Zmierz temperaturę
4. Wyświetlaj temperaturę
5. Wyłącz termometr
6. Konto bankowe posiada 26-cyfrowy numer nadawany podczas zakładania konta. Saldo początkowe konta wynosi 0 zł. Na konto możesz wpłacić dowolną kwotę. Możesz także wypłacić z konta dowolną kwotę, pod warunkiem, że nie przekracza ona salda konta. W przypadku próby wypłaty większej kwoty zostanie wyświetlony komunikat: „Za mało środków na koncie”. W każdej chwili możliwe jest wyświetlenie informacji o numerze i saldzie rachunku bankowego w następującym formacie:

Nr konta bankowego: 11 1111 1111 1111 1111 1111 1111  
Saldo: 25,38 zł

Stwórz program obsługujący konto bankowe.

1. Zapoznaje się z problemem.
2. Zidentyfikuj obiekt
3. Zdefiniuj stany i zachowania obiektu.
4. Przekształć stany i zachowania obiektu w pola i metody klasy, które posłużą jako wzorzec do tworzenia obiektu.
5. Utwórz szkic klasy bez tworzenia treści metody.
6. Utwórz treść każdej metody.

Następnie użyj programu i:

1. Załóż konto bankowe o numerze 12 3456 5555 9090 1111 0000 7722
2. Wyświetl saldo konta
3. Kaucja 25,30 zł
4. Wyświetl saldo konta
5. Wypłata 31,70 zł
6. Wyświetl saldo konta
7. Wypłata 14 zł
8. Wyświetl saldo konta
9. Napisz program zawierający klasę Statistics, która opisuje właściwości dowolnego zestawu liczb. Klasa powinna umożliwiać:
   1. Dodaj do zbioru liczb następną liczbę odczytaną z klawiatury (zapisz liczby w tablicy)
   2. Wyświetl wszystkie liczby oddzielone spacją
   3. Określ największą liczbę
   4. Określ najmniejszą liczbę
   5. Oblicz średnią arytmetyczną liczb
   6. Oblicz medianę
   7. Wyświetlanie obliczonych/określonych wielkości statystycznych (minimum, maksimum, średnia arytmetyczna, mediana)

Następnie użyj programu do liczb:

12, 37, 6, 9, 17

1. Klasa Kontakt zawiera pola 'nazwisko', 'e-mail' i 'telefon' umożliwiające opis pojedynczego kontaktu na smartfonie. Klasa Contact\_List umożliwia przechowywanie kontaktów (przechowywanie obiektów opisujących kontakty w tablicy) oraz wykonywanie następujących operacji:
   1. Dodaje nowy kontakt
   2. Wyświetla listę kontaktów

Napisz program składający się z 3 plików (smartphone.py, contact.py, contact\_list.py). W programie pocztowym (smartphone.py) utwórz obiekt reprezentujący listę kontaktów i dodaj następujące dane osób:

Jan Brown brown@onet.pl 555234000 Anna Maj             
    am@o2.pl 232000199 Jerzy Mały smallg@google.pl 222999100 Paola Duży bigpaola@poczta.pl 100200300             
       

Następnie wyświetl listę kontaktów dostępną na smartfonie.

1