

Programowanie Python 1

(CP1S02005)

Politechnika Białostocka - Wydział Elektryczny
Cyfryzacja przemysłu, sem. II, studia stacjonarne I stopnia
Rok akademicki 2023/2024

Wykład nr 7 (17.04.2024)

dr inż. Jarosław Forenc

Plan wykładu nr 7

- Przykładowe kolokwium - rozwiązanie
- Słownik
 - wczytywanie danych do słownika
 - zagnieżdżanie słownika w słowniku
- Macierze w Pythonie
 - implementacja w postaci listy list
 - odwołania do elementów
 - wybrane operacje

Python - kolokwium (zadanie 1, wersja 1)

- Umieść imię i nazwisko, nr grupy, nr albumu, datę, kod przedmiotu (CP1S02005), formę zajęć (Pracownia specjalistyczna) i nazwę środowiska programistycznego (Visual Studio Code) na początku kodu źródłowego programu (jako komentarz). [5 pkt.]

```
# Autor: Jan Kowalski, gr. PS1  
# Album: 123456  
# Data: 17-04-2024  
# Kod: CP1S02005  
# Forma: Pracownia specjalistyczna  
# IDE: Visual Studio Code
```

Python - kolokwium (zadanie 1, wersja 2)

- Umieść imię i nazwisko, nr grupy, nr albumu, datę, kod przedmiotu (CP1S02005), formę zajęć (Pracownia specjalistyczna) i nazwę środowiska programistycznego (Visual Studio Code) na początku kodu źródłowego programu (jako komentarz). [5 pkt.]

```
"""  
Autor: Jan Kowalski, gr. PS1  
Album: 123456  
Data: 17-04-2024  
Kod: CP1S02005  
Forma: Pracownia specjalistyczna  
IDE: Visual Studio Code  
"""
```

- zamiast trzech cudzysłowów można zastosować trzy apostrofy

Python - kolokwium (zadanie 2, wersja 1)

- Oblicz obwód elipsy L o parametrach c i d . W przypadku liczby π użyj odpowiedniej stałej. Wartości parametrów c i d wczytaj z klawiatury. Wyświetl wynik z dokładnością do 3 cyfr po kropce. [15 pkt.]

$$L = \pi(1,5(c + d) - \sqrt{cd})$$

```
import math
c = float(input("Podaj wartość parametru c: "))
d = float(input("Podaj wartość parametru d: "))
L = math.pi * (1.5 * (c + d) - math.sqrt(c * d))
print(f"Obwód elipsy wynosi: {L:.3f}")
```

```
Podaj wartość parametru c: 2.5
Podaj wartość parametru d: 5
Obwód elipsy wynosi: 24.236
```

Python - kolokwium (zadanie 2, wersja 2)

- Oblicz obwód elipsy L o parametrach c i d . W przypadku liczby π użyj odpowiedniej stałej. Wartości parametrów c i d wczytaj z klawiatury. Wyświetl wynik z dokładnością do 3 cyfr po kropce. [15 pkt.]

$$L = \pi(1,5(c + d) - \sqrt{cd})$$

```
import math
try:
    c = float(input("Podaj wartość parametru c: "))
    d = float(input("Podaj wartość parametru d: "))
except ValueError:
    print("Błędne dane wejściowe!")
else:
    L = math.pi * (1.5 * (c + d) - math.sqrt(c * d))
    print(f"Obwód elipsy wynosi: {L:.3f}")
```

Python - kolokwium (zadanie 3)

- Koszt ubezpieczenia samochodu stanowi określony procent jego wartości i jest zależny od wieku kierowcy. Kierowcy dzieleni są na cztery grupy (tabela).

Grupa	Wiek	Procent
0	od 18 do 25 lat	8,25%
1	powyżej 25 do 35 lat	7,25%
2	powyżej 35 do 45 lat	6,25%
3	powyżej 45 lat do 55 lat	6,00%
4	powyżej 55 lat	6,50%

- Wczytaj **wartość samochodu** oraz **wiek kierowcy**. Na podstawie wieku kierowcy, określ i wyświetl **grupę**, do której zostanie on zaliczony oraz oblicz i wyświetl **koszt** ubezpieczenia samochodu. Koszt wyświetl z dokładnością do 2 cyfr po kropce dziesiętnej, dodaj symbol waluty. **[15 pkt.]**

Python - kolokwium (zadanie 3)

```
wartość = float(input("Podaj wartość samochodu (w PLN): "))
wiek = int(input("Podaj wiek kierowcy: "))

if 18 <= wiek <= 25:
    grupa = 0
    koszt = 8.25 / 100 * wartość
elif 26 <= wiek <= 35:
    grupa = 1
    koszt = 7.25 / 100 * wartość
elif 36 <= wiek <= 45:
    grupa = 2
    koszt = 6.25 / 100 * wartość
elif 46 <= wiek <= 55:
    grupa = 3
    koszt = 6.00 / 100 * wartość
else:
    grupa = 4
    koszt = 6.50 / 100 * wartość

print(f"Grupa: {grupa}")
print(f"Koszt ubezpieczenia: {koszt:.2f} PLN")
```


Python - kolokwium (zadanie 3)

```
wartość = float(input("Podaj wartość samochodu (w PLN): "))
wiek = int(input("Podaj wiek kierowcy: "))

if 18 <= wiek <= 25:
    grupa = 0
    koszt = 8.25 / 100 * wartość
elif 26 <= wiek <= 35:
    grupa = 1
    koszt = 7.25 / 100 * wartość
elif 36 <= wiek <= 45:
    grupa = 2
    koszt = 6.25 / 100 * wartość
elif 46 <= wiek <= 55:
    grupa = 3
    koszt = 6.00 / 100 * wartość
else:
```

```
Podaj wartość samochodu (w PLN): 35000
Podaj wiek kierowcy: 23
Grupa: 0
Koszt ubezpieczenia: 2887.50 PLN
```

Python - kolokwium (zadanie 4)

- Utwórz listę zawierającą n liczb pseudolosowych z zakresu od 1 do 50. Wyświetl zawartość listy na ekranie. Wartość n wczytaj z klawiatury. [6 pkt.]
- Oblicz i wyświetl **średnią arytmetyczną** wszystkich liczb znajdujących się na liście. [4 pkt.]
- Utwórz dwie nowe listy. Do pierwszej skopiuj liczby **mniejsze lub równe** średniej, zaś do drugiej liczby **większe** od średniej. Wyświetl zawartość obu list na ekranie. [10 pkt.]

Suma punktów: [20 pkt.]

Python - kolokwium (zadanie 4)

```
import random

n = int(input("Podaj liczbę n: "))

liczby = [random.randint(1, 50) for _ in range(n)]

print("Liczby pseudolosowe:")
print(liczby)

srednia = sum(liczby) / n
print(f"Średnia arytmetyczna: {srednia:.2f}")

mniejsze = [nr for nr in liczby if nr <= srednia]
wieksze = [nr for nr in liczby if nr > srednia]

print("Liczby mniejsze lub równe średniej:")
print(mniejsze)

print("Liczby większe od średniej:")
print(wieksze)
```

Python - kolokwium (zadanie 4)

```
import random

n = int(input("Podaj liczbę n: "))

liczby = [random.randint(1, 50) for _ in range(n)]

print("Liczby pseudolosowe:")
print(liczby)

srednia = sum(liczby) / n
print(f"Średnia arytmetyczna: {srednia:.2f}")

mniejsze = [nr for nr in liczby if nr <= srednia]
```

```
Podaj liczbę n: 10
Liczby pseudolosowe:
[26, 46, 13, 18, 42, 28, 37, 3, 33, 17]
Średnia arytmetyczna: 26.30
Liczby mniejsze lub równe średniej:
[26, 13, 18, 3, 17]
Liczby większe od średniej:
[46, 42, 28, 37, 33]
```

Python - kolokwium (zadanie 5)

- Grupa osób odwiedziła w wakacje kilka miast w Europie. Zdefiniuj słownik, w którym klucze zawierają imiona osób, zaś wartości to lista miast, które odwiedziły te osoby. Jedna osoba mogła odwiedzić kilka miast. [4 pkt.]
- Zdefiniuj drugi słownik, w którym klucze zawierają nazwy miast, zaś wartości to odległość danego miasta od Białegostoku. Wykorzystaj poniższe miasta i odległości: Berlin - 659 km, Paryż - 1528 km, Londyn - 1588 km, Rzym - 1482 km, Praga - 692 km, Oslo - 1070 km. [4 pkt.]
- Dla każdej osoby wyświetl w jednym wierszu: imię osoby, odwiedzone miasta, sumę odległości z Białegostoku do tych miast. [10 pkt.]
- Wyświetl na ekranie imię osoby, dla której suma odległości jest największa. [7 pkt.]

Suma punktów: [25 pkt.]

Python - kolokwium (zadanie 5)

```
wyjazdy = {  
    "Kamil": ["Berlin", "Paryż", "Rzym"],  
    "Anna": ["Londyn", "Praga"],  
    "Ola": ["Oslo", "Berlin"],  
    "Szymon": ["Praga", "Rzym", "Berlin", "Londyn"],  
}  
odleglosci = {  
    "Berlin": 659,  
    "Paryż": 1528,  
    "Londyn": 1588,  
    "Rzym": 1482,  
    "Praga": 692,  
    "Oslo": 1070  
}
```

Python - kolokwium (zadanie 5)

```
max_osoba = ""
max_odleglosc = 0

for osoba, miasta in wyjazdy.items():
    tekst = osoba + ": "
    suma = 0

    for miasto in miasta:
        tekst += (miasto + ", ")
        suma += odleglosci[miasto]

    tekst += ("odległość: " + str(suma))
    print(tekst)

    if suma > max_odleglosc:
        max_odleglosc = suma
        max_osoba = osoba

print(f"Największa suma odległości: {max_osoba}")
```

Python - kolokwium (zadanie 5)

```
max_osoba = ""
max_odleglosc = 0

for osoba, miasta in wyjazdy.items():
    tekst = osoba + ": "
    suma = 0

    for miasto in miasta:
        tekst += (miasto + ", ")
        suma += odleglosci[miasto]

    tekst += ("odległość: " + str(suma))
    print(tekst)

    if suma > max_odleglosc:
```

```
Kamil: Berlin, Paryż, Rzym, odległość: 3669
Anna: Londyn, Praga, odległość: 2280
Ola: Oslo, Berlin, odległość: 1729
Szymon: Praga, Rzym, Berlin, Londyn, odległość: 4421
Największa suma odległości: Szymon
```


Python - kolokwium (zadanie 6)

- Trzy listy zawierają imiona osób, które brały udział w trzech rundach zawodów sportowych. Te same osoby mogą występować na kilku listach. [6 pkt.]
- Wyświetl na ekranie alfabetyczną listę unikalnych osób, które wzięły udział przynajmniej w jednej rundzie zawodów. Imiona nie mogą się powtarzać. [14 pkt.]

Suma punktów: [20 pkt.]

Python - kolokwium (zadanie 6, wersja 1)

```
zawody1 = ["Anna", "Piotr", "Jan", "Ela", "Marcin"]
zawody2 = ["Jan", "Ola", "Paweł", "Ela"]
zawody3 = ["Marcin", "Jan", "Kamila", "Mariusz"]

wszyscy = set()

wszyscy.update(zawody1)
wszyscy.update(zawody2)
wszyscy.update(zawody3)

wszyscy = sorted(wszyscy)

print("Lista uczestników:")
for osoba in wszyscy:
    print(osoba)
```

```
Lista uczestników:
Anna
Ela
Jan
Kamila
Marcin
Mariusz
Ola
Paweł
Piotr
```

Python - kolokwium (zadanie 6, wersja 2)

```
zawody1 = ["Anna", "Piotr", "Jan", "Ela", "Marcin"]
zawody2 = ["Jan", "Ola", "Paweł", "Ela"]
zawody3 = ["Marcin", "Jan", "Kamila", "Mariusz"]

wszyscy = sorted(set(zawody1 + zawody2 + zawody3))

print("Lista uczestników:")
for osoba in wszyscy:
    print(osoba)
```

```
Lista uczestników:
Anna
Ela
Jan
Kamila
Marcin
Mariusz
Ola
Paweł
Piotr
```

Python - słownik (wczytywanie danych)

- wczytywanie danych z klawiatury bezpośrednio do słownika

```
kierowcy = {}
wczytuj = True

while wczytuj:
    imię = input("Podaj imię kierowcy: ")
    marka = input("Podaj markę samochodu: ")
    kierowcy[imię] = marka

    odp = input("Dalej? (tak/nie): ")
    if odp == "nie":
        wczytuj = False

print("Dane: ")
for imię, marka in kierowcy.items():
    print(f"{imię} ma samochód {marka}")
```

Python - słownik (wczytywanie danych)

- wczytywanie danych z klawiatury bezpośrednio do słownika

```
kierowcy = {}  
wczytuj = True  
  
while wczytuj:  
    imię = input("Podaj imię  
    marka = input("Podaj markę  
    kierowcy[imię] = marka  
  
    odp = input("Dalej? (tak/nie):  
    if odp == "nie":  
        wczytuj = False  
  
print("Dane: ")  
for imię, marka in kierowcy.items():  
    print(f"{imię} ma samochód {marka}")
```

```
Podaj imię kierowcy: Jan  
Podaj markę samochodu: Fiat  
Dalej? (tak/nie): tak  
Podaj imię kierowcy: Anna  
Podaj markę samochodu: Skoda  
Dalej? (tak/nie): tak  
Podaj imię kierowcy: Piotr  
Podaj markę samochodu: BMW  
Dalej? (tak/nie): nie  
Dane:  
Jan ma samochód Fiat  
Anna ma samochód Skoda  
Piotr ma samochód BMW
```

Python - słownik (zagnieżdżanie)

- zagnieżdżanie słownika w słowniku

```
student = {  
    "imię" : "Jan",  
    "nazwisko" : "Kowalski",  
    "oceny" : {  
        "Metrologia" : 4.5,  
        "Fizyka" : 3.0,  
        "Matematyka" : 4.0  
    }  
}  
  
print(f"Imię: {student["imię"]}")  
print(f"Nazwisko: {student["nazwisko"]}")  
  
print("Oceny:")  
for przedmiot, ocena in student["oceny"].items():  
    print(f"{przedmiot}: {ocena}")
```

```
Imię: Jan  
Nazwisko: Kowalski  
Oceny:  
Metrologia: 4.5  
Fizyka: 3.0  
Matematyka: 4.0
```

Python - macierze

- do zaimplementowania macierzy można użyć **list zagnieżdżonych** lub metod z modułu **NumPy**
- korzystanie z biblioteki **NumPy** zalecane jest w przypadku bardziej zaawansowanych operacji algebraicznych na macierzach
- w przypadku **zagnieżdżenia list**, każdy wiersz macierzy jest listą, a cała macierz jest listą list

```
macierz = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
]  
  
print("Macierz:")  
for wiersz in macierz:  
    print(wiersz)
```

```
Macierz:  
[1, 2, 3]  
[4, 5, 6]  
[7, 8, 9]
```

Python - macierze (odwołania do elementów)

- dostęp do elementu znajdującego się na pozycji (wiersz, kolumna), wymaga zastosowania indeksowania dwukrotnego `[wiersz][kolumna]`

```
macierz = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
]  
element = macierz[1][2]  
print("Element:", element)
```

Element: 6

- można również użyć pętli zagnieżdżonych, aby przejść przez wszystkie elementy macierzy

```
for wiersz in macierz:  
    for element in wiersz:  
        print(element)
```

1
2
...

Python - macierze (wybrane operacje)

```
import random

# Generowanie i wyświetlanie macierzy o rozmiarze 3x4
macierz = []
for _ in range(3):
    wiersz = [random.randint(0, 99) for _ in range(4)]
    macierz.append(wiersz)

print("Macierz:")
for wiersz in macierz:
    print(wiersz)
```

```
Macierz:
[86, 5, 15, 74]
[37, 19, 78, 27]
[47, 22, 8, 92]
```

Python - macierze (wybrane operacje)

```
# Obliczanie sumy elementów macierzy
suma = 0
for wiersz in macierz:
    suma += sum(wiersz)
print("Suma elementów macierzy:", suma)

# Obliczanie średniej arytmetycznej elementów macierzy
ilosc_elementow = 0
suma = 0
for wiersz in macierz:
    ilosc_elementow += len(wiersz)
    suma += sum(wiersz)
srednia = suma / ilosc_elementow
print("Średnia arytmetyczna elementów macierzy:", srednia)
```

```
Suma elementów macierzy: 510
Średnia arytmetyczna elementów macierzy: 42.5
```

Python - macierze (wybrane operacje)

```
# Znajdowanie wartości min i max w macierzy
min_wartosc = float('inf')
max_wartosc = float('-inf')
for wiersz in macierz:
    min_wartosc = min(min_wartosc, min(wiersz))
    max_wartosc = max(max_wartosc, max(wiersz))
print("Wartość najmniejsza w macierzy:", min_wartosc)
print("Wartość największa w macierzy:", max_wartosc)
```

```
Wartość najmniejsza w macierzy: 5
Wartość największa w macierzy: 82
```

Koniec wykładu nr 7

Dziękuję za uwagę!