

**I Zagadnienia do opanowania**

1. Co to jest lista, krotka, słownik i zbiór?
2. Jak się definiuje i wykorzystuje te obiekty?
3. Wyjątki i ich obsługa.
4. Operacje na plikach. Różnice między obsługą plików tekstowych, a binarnych.
5. Serializacja i deserializacja danych przy zapisie do i odczycie z pliku binarnego (moduł *pickle*).

**II Rozgrzewka...**1. Krotki:

a) Jaki będzie wynik działania poniższego skryptu?

```
k = (1, 2.4, 'ABC', 3-2j, (1,3))
for i,j in enumerate(k):
    print(str(i)+':',j)
print('pierwszy element:',k[0])
print('ostatni element:',k[-1])
```

b) Krotki mogą być zagnieżdżone (bo mogą zawierać m.in. krotki). Jaki będzie zatem wynik działania poniższego skryptu?

```
k = (((('A','B'), (1,2)), (7.0, 6e2))
print(k[0])
print(k[-1])
print(k[0][0][1])
print(k[1][1])
```

2. Listy:

a) Jaki będzie wynik następującego skryptu?:

```
1 imiona=list()
2 imiel = 'Paweł'
3 imie2 = 'Kasia'
4 imiona += imiel
5 imiona += imie2
6 w=len(imiona)
7 print('Długość listy imion:',w)
8 print('Zawartość listy imion:',imiona)
```

b) ... a jaki, gdy zmodyfikujemy linie 4 i 5 kodu, w następujący sposób?:

```
imiona += [imiel]
imiona += [imie2]
```

c) Jakie znasz inne metody dodawania elementów do listy oraz usuwania elementów z listy? Zmodyfikuj powyższy kod, wykorzystując te metody (dodawania i usuwania elementów).

3. Słowniki:

a) Jaki będzie wynik poniższych działań?:

```
>>> s = {'Imie': 'Józef', 'Nazwisko': 'Nowak', 5: 13.2}
>>> s.keys()

>>> s.values()

>>> s.items()

>>> w = s.get('Imie')
>>> w

>>> s

>>> w = s.pop(5)
>>> w

>>> s
```

b) Jak dopisuje się dane do słownika? Dopisz do słownika *s* trzy różne pary klucz : wartość.4. Zbiory:

a) Jaki będzie wynik poniższych operacji?:

```
>>> x = {'jeden', 'dwa', 'trzy'}
>>> y = {'dwa', 'trzy', 'cztery'}
>>> x.intersection(y)

>>> {'jeden', 'trzy'}.issubset(x)

>>> {'jeden', 'trzy'}.issubset(y)

>>> x|y

>>> x&y

>>> x^y

>>> x-y

>>> z={1,2,3,1,1,2,2,3,3,1,2,3,1,3,2,1,2,3}
>>> print('Długość: ', len(z), 'Zbiór: ', z)
```

b) Jak wykorzystać zbiór do usuwania duplikatów z listy? Pokaż na przykładzie.

## 5. Obsługa wyjątków:

a) Przetestuj poniższy skrypt. Co będzie wynikiem jego działania gdy użytkownik poda żadaną liczbę całkowitą, a co jeśli tego nie zrobi?

```
try:
    x = int(input('Podaj liczbę całkowitą: '))
except ValueError:
    print('To nie jest liczba całkowita!')
else:
    print('Wprowadziłeś liczbę', x)
finally:
    print('Dziękuję!')
```

b) Przyjrzyj się drzewu hierarchii klas wyjątków wbudowanych (built-in exceptions) dostępnej m.in. tu: <http://docs.python.org/3.2/library/exceptions.html#exception-hierarchy>. Zastanów się, dla ilu z tych wyjątków jesteś w stanie, bez sprawdzania w pomocy podręcznej, określić czego dotyczą i kiedy mogą być zgłaszane (rzucane)?

c) Rozbuduj powyższy przykład z pkt. a w taki sposób, aby od użytkownika były pobierane dwie liczby całkowite i zwracany był iloraz tych liczb. Przechwyć i obsłuż ewentualne wyjątki.

## 6. Operacje na plikach:

a) Jaki będzie wynik poniższych działań? Co oznaczają poszczególne linie kodu?

```
>>> f = open('plik1.txt', 'wt+')
>>> f.write('ABCDEFGHJKLMNOP')

>>> f.tell()

>>> f.seek(0)

>>> f.read(6)

>>> f.read()

>>> f.write('''pierwszy
...         drugi
...         trzeci''')

>>> f.seek(0)
>>> f.readline()

>>> f.readline()

>>> f.seek(0)
>>> f.readlines()

>>> if not f.closed: f.close()
```

b) Sprawdź poniższy kod. Czy do pliku plik2.txt został zapisany tekst 'próba zapisu'? Czy plik został automatycznie zamknięty?

```
>>> with open('plik2.txt', 'wt') as f:
...     f.write('próba zapisu')

>>> f.closed
```

### III Zadania do rozwiązania:

1. Napisz skrypt, który będzie pobierał od użytkownika cyfrę i wyświetlał ją słownie, np. 1 → jeden, 2 → dwa itd. (wykorzystaj w tym celu słownik!).
2. Napisz skrypt, który będzie pobierał od użytkownika w pętli (np. 10 razy) imię i dopisywał je do listy imion. Sprawdź czy użytkownik wpisuje tylko litery (imiona zwykle składają się wyłącznie z liter :) ), a przed zapisem imienia do listy odpowiednio je sformatuj (imiona piszemy wielką literą!). Ponadto, skrypt powinien sprawdzać, czy wprowadzone imię nie znajduje się już na liście. Jeśli jest na liście, to powiadom o tym użytkownika. Na koniec, niech zostanie wyświetlona cała lista zapisanych imion.
3. Zmodyfikuj skrypt z pkt. 2 tak, żeby lista imion była uzupełniana aż do momentu, gdy użytkownik zamiast imienia wpisze hasło np. 'koniec' (niezależnie od wielkości liter). Wówczas lista imion zostanie posortowana rosnąco, a następnie zostanie wyświetlona na ekranie i zapisana do pliku tekstowego (np. imiona.txt) w taki sposób, żeby każde imię z listy było w osobnej linii pliku.
4. Zmodyfikuj skrypt z pkt. 3 tak, żeby lista imion była inicjalizowana poprzez wczytywanie jej z pliku tekstowego (np. imiona.txt).
5. Uzupełnij powyższy skrypt o rozpoznawanie polskich imion żeńskich i męskich (w języku polskim, wszystkie imiona żeńskie kończą się na literę 'a', a wśród imion męskich tylko: Kosma, Barnaba, Bonawentura, Jarema i Kuba). W zależności od tego, czy imię jest męskie, czy żeńskie, będzie zapisywane do odpowiedniej listy imion. Obie listy powinny zostać posortowane i wyświetlone, a następnie zapisane do pliku binarnego (wykorzystaj serializację – czyli moduł *pickle*).
6. Ostatnia modyfikacja... czyli inicjalizacja obu list imion (imiona męskie i żeńskie) poprzez odczyt z pliku binarnego (deserializacja!).