

# Ćwiczenie 4

## Operacje na łańcuchach znaków, pliki i wyjątki

### 1 Wprowadzenie

#### 1.1 Klasa `str`

Klasa `str` dostarcza wysokopoziomowe narzędzia przetwarzania tekstów:

<https://docs.python.org/3.5/library/stdtypes.html#text-sequence-type-str>

Wybrane metody:

Nazwa	Opis
<code>find</code>	Wyszukiwanie pierwszego wystąpienia w łańcuchu podanego podłańcucha znaków
<code>lower</code>	Zamiana wszystkich liter w łańcuchu na małe litery
<code>split</code>	Podział łańcucha względem podanego separatora
<code>startswith</code>	Sprawdzenie czy łańcuch zaczyna się od podanego prefiksu
<code>upper</code>	Zamiana wszystkich liter w łańcuchu na duże litery

#### 1.2 Pliki

- Otwarcie wejściowego pliku tekstowego do odczytu:

```
obiekt_pliku=open(plik, 'r')
```

Wybrane metody:

Nazwa	Opis
<code>read</code>	Odczyt całej zawartości pliku do pojedynczego łańcucha znaków
<code>readline</code>	Odczyt jednego wiersza pliku
<code>readlines</code>	Odczyt całej zawartości pliku do listy łańcuchów znaków odpowiadających kolejnym wierszom

- Otwarcie wyjściowego pliku tekstowego do zapisu:

```
obiekt_pliku=open(plik, 'w')
```

Otwarcie wyjściowego pliku tekstowego do dopisywania:

```
obiekt_pliku=open(plik, 'a')
```

Wybrane metody:

Nazwa	Opis
<i>write</i>	Zapis łańcucha znaków do pliku
<i>writelines</i>	Zapis listy łańcuchów znaków do pliku

- Zamknięcie pliku:

```
obiekt_pliku.close()
```

### 1.3 Wyjątki

- Wybrane klasy:

Nazwa	Opis
<i>BaseException</i>	Główna klasa bazowa dla wyjątków wbudowanych
<i>Exception</i>	Klasa bazowa dla wszystkich wyjątków, które nie powodują zakończenia działania systemu, także dla wyjątków definiowanych przez użytkownika
<i>IOError</i>	Klasa wyjątku zgłaszanego podczas niepowodzenia operacji wejścia/wyjścia
<i>IndexError</i>	Klasa wyjątku zgłaszanego podczas odwoływania się poza zakres sekwencji

- Obsługa wyjątków:

```
try:
    # ...kod nadzorowany...
except KlasaWyjątku:
    # ...kod wykonywany przy wystąpieniu wyjątku...
else:
    # ...kod wykonywany gdy nie wystąpił wyjątek...
finally:
    # ...kod zawsze wykonywany...
```

- Człony *else* i *finally* są opcjonalne.
- Może wystąpić więcej członów *except*.

- Tworzenie własnego wyjątku:

```
class nazwaKlasyWyjątku(Exception):
    # ...ciało klasy...
```

- Zgłaszanie wyjątku:

```
raise nazwaKlasyWyjątku()
```

## 2 Zadania

### 2.1

Napisz skrypt umożliwiający:

- pobieranie i zapisywanie danych o obiektach klasy *KlientBanku* opisywanych przez:

- nazwisko,
- imię,
- numer konta,
- stan konta,

(dane o obiektach powinny być przechowywane w pliku tekstowym),

- wyszukiwanie klientów banku po wpisaniu dowolnej liczby początkowych liter nazwiska.

W skrypcie powinny być obsługiwane wyjątki:

- wyjątek wbudowany generowany w przypadku problemów z otwarciem lub zapisem pliku,
- własny wyjątek generowany w przypadku błędnego formatu pliku.

## 2.2

Napisz skrypt umożliwiający:

- pobieranie i zapisywanie danych o obiektach klasy *Druzyna* opisywanych przez:
  - nazwę (mogącą składać się z kilku członów oddzielonych spacjami),
  - liczbę meczy,
  - liczbę punktów,

(dane o obiektach powinny być przechowywane w pliku tekstowym),

- wyświetlanie tylko pierwszych członów nazw drużyn.

W skrypcie powinny być obsługiwane wyjątki:

- wyjątek wbudowany generowany w przypadku problemów z otwarciem lub zapisem pliku,
- własny wyjątek generowany w przypadku błędnego formatu pliku.