

Programowanie I

Zajęcia nr 7

Rafał Masełek

21. kwietnia 2022r.

Oryginalne zadania przygotowane przez B. Zglinickiego.

Lista wbudowanych wyjątków:

<https://docs.python.org/3/library/exceptions.html>

<https://www.programiz.com/python-programming/exceptions>

Własny wyjątek

```
# własna definicja wyjątku
class MyError(Exception):
    # Constructor or Initializer
    def __init__(self, value):
        self.value = value

    # __str__ is to print() the value
    def __str__(self):
        return(repr(self.value))
```

Zadanie 1. comments – Usuwanie komentarzy

Napisz program `comments`, służący do usuwania komentarzy w pliku tekstowym. Program przyjmuje trzy argumenty wywołania: pierwszy jest pewnym znakiem, zaś pozostałe dwa to nazwy plików. Program przepisuje zawartość pierwszego z tych plików do drugiego z nich, pomijając linie zaczynające się znakiem przekazany programowi jako pierwszy argument.

Wykorzystując mechanizm wyjątków zabezpiecz program przed sytuacjami, gdy pierwszy argument jego wywołania nie jest pojedynczym znakiem – stwórz w tym celu własny wyjątek, oraz gdy plik o nazwie podanej jako drugi argument wywołania nie istnieje.

Przykładowe wykonanie

Zawartość pliku wejściowego `in.txt`:

```
    pierwsza linia
    ?komentarz
    druga linia
```

Wywołanie:

```
py comments ? in.txt out.txt
```

Zawartość pliku wyjściowego `out.txt`:

```
    pierwsza linia
    druga linia
```

Zadanie 2. `textcalc` – Kalkulator tekstowy

Napisz funkcję `textcalc`, która przyjmuje jako argument łańcuch tekstowy przedstawiający działanie matematyczne, posiadający następującą strukturę:

- ciąg znaków dający się zinterpretować jako liczba rzeczywista (składający się z cyfr i ewentualnie jednej kropki),
- spacja,
- jeden z następujących znaków: `+`, `-`, `*`, `/`, `%`, `^`,
- spacja,
- ciąg znaków dający się zinterpretować jako liczba rzeczywista (składający się z cyfr i ewentualnie jednej kropki).

Wymienione znaki symbolizują operacje matematyczne, odpowiednio: dodawanie, odejmowanie, mnożenie, dzielenie, obliczanie reszty z dzielenia oraz potęgowanie. Funkcja zwraca liczbę rzeczywistą będącą wynikiem operacji matematycznej powiązanej z danym znakiem, wykonanej na zadanych liczbach. Na przykład dla argumentu `2.5 + 1.3` funkcja powinna zwrócić wartość `3.8`. W przypadku, gdy wartość argumentu jest niepoprawna (czyli nie ma odpowiedniej struktury), funkcja powinna zgłosić wyjątek.

Korzystając z tej funkcji, napisz program `textcalc`, będący prostym kalkulatorem tekstowym. Program przyjmuje jako argumenty wywołania nazwy dwóch plików tekstowych. Pierwszy z nich zawiera ciąg działań matematycznych zapisanych w sposób opisany powyżej, taki sam, jak dla argumentu funkcji `textcalc`. Każde działanie zapisane jest w osobnej linii, ilość działań nie jest znana z góry. Program przepisuje zawartość tego pliku do drugiego pliku, dopisując na końcu każdej linii spację, znak równości, kolejną spację i wynik działania zapisanego w tej linii. Wykorzystując mechanizm wyjątków, zabezpiecz program przed możliwymi sytuacjami wyjątkowymi.

Przykładowe wykonanie 1.

Zawartość pliku wejściowego `in.txt`:

```
3.67 + 2
3.2 * 4
3.14 / 2.72
```

Wywołanie:

```
py textcalc in.txt out.txt
```

Zawartość pliku wyjściowego `out.txt`:

```
3.67 + 2 = 5.67
3.2 * 4 = 12.8
3.14 / 2.72 = 1.15
```

Przykładowe wykonanie 2.

Zawartość pliku wejściowego `in.txt`:

```
3.67 + 2
3.2 * żółw
3.14 / 2.72
```

Wywołanie:

```
py textcalc in.txt out.txt
```

Zawartość pliku wyjściowego `out.txt`:

```
3.67 + 2 = 5.67
Błąd! "żółw" nie jest liczbą.
3.14 / 2.72 = 1.15
```

Zadanie 3 enigma – Szyfrowanie z użyciem XOR

Jedną z metod szyfrowania plików tekstowych jest zastąpienie każdego znaku bitową różnicą symetryczną (*XOR*) jego kodu ASCII zadaną jednobajtową liczbą, zwaną kluczem. Odszyfrowanie tekstu polega na ponownym obliczeniu bitowej różnicy symetrycznej zaszyfrowanych kodów z tym samym kluczem.

Napisz program `enigma`, szyfrujący i deszyfrujący pliki tekstowe omówioną metodą. Program powinien przyjmować cztery argumenty wywołania. Pierwszy z nich to jednobajtowa liczba całkowita będąca kluczem, drugi to `/e` dla szyfrowania lub `/d` dla deszyfrowania, trzeci jest nazwą pliku z tekstem do zaszyfrowania/odszyfrowania, zaś czwarty – nazwą pliku, w którym ma zostać zapisany zaszyfrowany/odszyfrowany tekst. Szyfrowanie odbywa się znak po znaku, włączając znaki białe. Plik z zaszyfrowanym tekstem powinien zawierać oddzielone spacjami liczby – każda z nich jest bitową różnicą symetryczną kodu zaszyfrowanego znaku i klucza. Wykorzystując mechanizm wyjątków, zabezpiecz program przed możliwymi sytuacjami wyjątkowymi.

Przykładowe wykonanie 1.

Zawartość pliku wejściowego `in.txt`:

```
Non intratur in veritate, nisi per caritate.
```

Wywołanie:

Windows: `enigma.exe 7 /e in.txt out.txt`

Linux: `./enigma 7 /e in.txt out.txt`

Zawartość pliku wyjściowego out.txt:

```
73 104 105 39 110 105 115 117 102 115 114 117 39 110 105 39 113 98
117 110 115 102 115 98 106 43 39 105 110 116 110 39 119 98 117 39
100 102 117 110 115 102 115 98 106 41
```

Przykładowe wykonanie 2.

Zawartość pliku wejściowego in.txt:

```
73 104 105 39 110 105 115 117 102 115 114 117 39 110 105 39 113 98
117 110 115 102 115 98 106 43 39 105 110 116 110 39 119 98 117 39
100 102 117 110 115 102 115 98 106 41
```

Wywołanie:

Windows: `enigma.exe 7 /d in.txt out.txt`

Linux: `./enigma 7 /d in.txt out.txt`

Zawartość pliku wyjściowego out.txt:

Non intratur in veritatem, nisi per caritatem.