Nazwa implementacji: Nauka języka Python - wyjątki Autor: Piotr Fiorek **Opis implementacji:** Poznanie czym są wyjątki i jak ich używać.

Do tej pory zawsze zakładaliśmy, że nasz kod działa poprawnie. Co się jednak dzieje kiedy coś pójdzie nie tak? W takich sytuacjach wyrzucany jest wyjątek. Wyjątek jest obiektem specjalnego typu, który powoduje awaryjne przerwanie wykonania programu. Wyjątek jest wyrzucany np. kiedy staramy się odwołać do nieistniejącego elementu listy:

```
>>> a = [1, 2, 3]

>>> a[2]

3

>>> a[3]

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

IndexError: list index out of range
```

To, co widać to informacja o tym, że coś poszło nie tak i szczegółowe dane o tym, co dokładnie się nie udało. Pierwsza linia mówi, że wystąpił błąd. Druga (w tym wypadku tylko jedna linia, ale czasami jest ich więcej) informuje, gdzie program napotkał problem – w tym wypadku w pierwszej linii, bo tylko tyle wprowadziliśmy do wykonania. Ostatnia linia informuje, jakiego rodzaju błąd wystąpił. „IndexError” jest typem wyjątku, który został wyrzucony i jak nietrudno zgadnąć oznacza, że numer indeksu, do którego próbujemy się odwołać, jest niepoprawny. Po dwukropku następuje słowny opis błędu, który mówi coś więcej, czyli w tym przypadku informujący, że podaliśmy zbyt dużą cyfrę jako indeks listy.

Wyrzucony wyjątek może zostać przez program złapany i obsłużony. Kiedy wyjątek jest wyrzucany, wykonanie programu jest przerywane i wyjątek jest wyrzucany wyżej (czyli do funkcji, która wywołała funkcję, która wyrzuciła wyjątek) tak długo, aż zostanie obsłużony lub dopóki nie będzie już nic powyżej i wtedy program kończy swoje działanie z błędem. Wyjątki obsługuje się specjalną składnią, która wygląda tak:

```
try:

instrukcja1

instrukcja2

...

except:

instrukcja1

instrukcja2

...
```

W praktyce wygląda to tak:

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
>>> a = [1, 2, 3]

>>> try:

... a[3]

... except:

... print('poza zakresem listy')

...
```

poza zakresem listy

W powyższym kodzie deklarujemy listę i w sekcji dla **„podejrzanych instrukcji”** (po „**try**”) próbujemy odwołać się do trzeciego elementy tej listy. Następnie po instrukcji **„**

except

” następuje kod, który będzie wywołany w momencie wyrzucenia wyjątku. Powyższy kod można przekształcić na funkcję, która będzie przyjmowała tablice i numer indeksu z tej tablicy i będzie próbowała wyświetlić wartość spod tego indeksu. Jeśli natomiast numer indeksu będzie nieprawidłowy, będzie zwracała ostatni element z tablicy:

```
>>> def wyswietl(tablica, indeks):

... try:

... print(tablica[indeks])

... except:

... print(tablica[len(tablica)-1])

...

>>> a = [1, 2, 3]

>>> wyswietl(a, 0)

1

>>> wyswietl(a, 2)

3

>>> wyswietl(a, 3)

3

>>> wyswietl(a, 7)
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

3

Taki zapis „try ... except ...” wyłapie wszystkie wyjątki, jakie instrukcja po „try” wyrzuci. Oczywiście w naszym przypadku nie ma zbyt wielu wyjątków, które mogą zostać wyrzucone. Jednak w bardziej złożonym kodzie, gdzie po „try” jest nie jedna, a kilka instrukcji, kilka różnych wyjątków może zostać wyrzuconych i dobrze jest rozróżniać, który wyjątek został wyrzucony i różnie je obsługiwać. W tym celu po słowie „

except

”, a przed dwukropkiem, wpisujemy typ wyjątku, jaki ma być obsłużony. W naszym przypadku będzie to „IndexError”:

```
>>> def wyswietl(tablica, indeks):
... try:
... print(tablica[indeks])
... except IndexError:
... print(tablica[len(tablica)-1])
```

Tak wygląda elegancko zapisana nasza funkcja. W tym przypadku nie robi to zbyt dużej różnicy, ale warto się uczyć dobrych wzorców. W prawdziwym kodzie może się zdarzyć, że będzie nam się wydawało, że dany fragment może wyrzucić tylko jeden rodzaj wyjątku, więc nie wpiszemy jego typu, pozostawiając samo słowo **„except”**, a gdy potem będziemy się zastawiać czemu program nie działa, może się okazać, że kod wyrzuca jeszcze inny rodzaj wyjątku, który zostaje wyłapany i co gorsza źle obsłużony.

Konstrukcja

„try ... except ...”

może mieć na końcu dołożoną opcjonalną część

„else”

, która będzie wywoływana, jeśli kod wewnątrz sekcji

„try”

zostanie wykonany poprawnie bez wyrzucania wyjątku.

Poza **„else”** jest jeszcze, również opcjonalna, część **„finally”**, która z kolei jest wykonywana zawsze, jeśli tylko została dopisana. Jest ona najczęściej używana do **„posprzątania”** po kodzie, który znajduje się w sekcji **„try”**.

Zadania

Napisz program, w którym użytkownik najpierw wpisuje elementy, które są dodawane do tablicy, a potem użytkownik wybiera, który element wyświetlić. Jeśli element o zadanym numerze nie istnieje, program wyrzuci wyjątek, który ma być obsłużony.

Zmodyfikuj kalkulator z zajęć tak, aby wyjątek dzielenia przez zero był obsługiwany.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.