Indeksowanie i wycinanie — __getitem__ i __setitem__

Jeśli w klasie jest zdefiniowana (lub dziedziczona po klasie nadrzędnej) metoda `__getitem__`, będzie ona automatycznie użyta w przypadkach prób wydobycia elementów po indeksach. Na przykład jeśli klasa `X` wystąpi w kontekście indeksowania `X[i]`, Python wywoła jej metodę `__getitem__`, przekazując `X` w pierwszym argumencie, a indeks w drugim. Poniższa klasa zwraca kwadrat wartości indeksu:

```
>>> class Indexer:
...     def __getitem__(self, index):
...         return index ** 2
...
>>> X = Indexer()
>>> X[2]                                # X[i] wywołuje X.__getitem__(i)
4

>>> for i in range(5):
...     print(X[i], end=' ')            # Przy każdej iteracji wywołuje __getitem__(X, i)
...
0 1 4 9 16
```

Wycinki

Co interesujące, metoda `__getitem__` jest wywoływana również w *wyrażeniach* wycinania. Wbudowane typy obsługują wycinanie w ten sam sposób. Poniższy listing przedstawia operację wycinania przy wykorzystaniu dolnego i górnego zakresu oraz argumentu przesunięcia (więcej informacji na temat operacji wycinania można znaleźć w rozdziale 7.).

```
>>> L = [5, 6, 7, 8, 9]
>>> L[2:4]                              # Wycinanie z użyciem składni wycinków
[7, 8]
>>> L[1:]
[6, 7, 8, 9]
>>> L[:-1]
[5, 6, 7, 8]
>>> L[::2]
[5, 7, 9]
```

W rzeczywistości parametry wycinania są pakowane w specjalny *obiekt wycinka*, który jest przekazywany do instancji listy. Obiekt wycinka można przekazać ręcznie: składanie wycinków to jedynie składniowy skrót, ułatwiający pracę z obiektami wycinków.

```
>>> L[slice(2, 4)]                      # Wycinanie z użyciem obiektów
[7, 8]
>>> L[slice(1, None)]
[6, 7, 8, 9]
>>> L[slice(None, -1)]
[5, 6, 7, 8]
>>> L[slice(None, None, 2)]
[5, 7, 9]
```

Ta obserwacja ma znaczenie dla klas implementujących metodę `__getitem__`, która jest wywoływana przy operacji indeksowania (wówczas otrzyma ona liczbę całkowitą) oraz wycinania (otrzyma obiekt wycinka). Nasz poprzedni przykład nie obsłuży wycinania, ponieważ metoda `__getitem__` zakłada, że otrzymała liczbę całkowitą, co naprawiamy w poniższym przykładzie. W przypadku indeksowania argument metody `__getitem__` jest liczbą całkowitą, jak poprzednio: