

Vector

Co to jest Vector?

Vector jest klasą w języku Java pozwalającą na przechowywanie obiektu dowolnego typu w postaci listy, którą można swobodnie modyfikować. Klasa Vector jest pochodną bardziej ogólnej klasy List, reprezentującej różnego rodzaju listy i faktycznie przeznaczona jest do tworzenia obiektów służących do przechowywania łatwo modyfikowalnej listy innych obiektów. Obiekty klasy Vector <T> obsługują tzw. typy generyczne - definiowane są w sposób niezależny od klasy obiektu który będą reprezentować, dzięki czemu mogą być wykorzystywane do tworzenia list obiektów dowolnych klas.

Deklarowanie obiektów klasy Vector

Aby przechować dane za pomocą obiektu klasy Vector należy go zadeklarować.

```
Vector<String> v = new Vector<>();
```

Powyższy zapis tylko pozornie wydaje się skomplikowany. Oznacza po prostu deklarację referencji do obiektu klasy Vector przeznaczonego do przechowywania obiektów klasy String, czyli po prostu tekstów. Po utworzeniu referencji w następnej kolejności stworzymy obiekt klasy Vector, czyli wywołujemy jego konstruktor, który w tym przypadku ma trochę nietypową składnię, ze względu na obecność ostrych nawisów <> . Podobnie jak wcześniej używamy jej na zadeklarowanie faktu, że utworzony obiekt klasy Vector będzie służył do przechowywania tekstu.

Praktyczne metody klasy Vector

add(String s)

Korzystanie z utworzonego obiektu klasy Vector jest stosunkowo proste. Po prostu dodajemy kolejne elementy do listy lub usuwamy elementy niepotrzebne. Realizujemy to za pomocą prostych poleceń.

```
v.add("Co masz zrobić dziś - zrób dziś.");  
v.add("Co się odwlecze to nie będzie szybciej.");  
v.add("Szczęście pomaga tylko tym, którzy pomagają sobie.");  
v.add("Gdyby nie wyjątki, zasady byłyby nie do zniesienia.");  
v.add("Podróż tysiąca mil zawsze zaczyna się od jednego kroku.");
```

W ten sposób utworzyłeś listę pięciu wróżb.

Oczywiście kolejne elementy można dodawać do listy z dowolnego źródła - na przykład z

konsoli.

```
Scanner sc = new Scanner(System.in);  
  
System.out.println("Podaj wróżbę do dopisania do listy: ")  
  
v.add(sc.next());
```

Kolejne elementy zostały umieszczone na końcu dotychczasowej listy elementów. Możliwe jednak jest dodawanie nowych elementów w dowolnym miejscu listy reprezentowanej przez obiekt klasy *Vector*.

add(int idx, String s)

Dzięki parametrowi *idx* można określić na której pozycji znajdzie się nowy wstawiany element listy

```
v.add(3, "Gdyby nie wyjątki, zasady byłyby nie do zniesienia.");
```

size()

Metoda umożliwiająca ustalenie długości listy.

remove(int idx)

Równie proste jak dodawanie elementów do listy jest ich usuwanie. Aby usunąć element z listy należy użyć metody *remove(int idx)*, gdzie *idx* jest numerem elementu do usunięcia. Po usunięciu elementu z listy na pozycji *idx* jest ona automatycznie skracana, a istniejące elementy znajdujące się wcześniej na prawo od usuniętego elementu są przesuwane o jedną pozycję w prawo.

```
v.remove(3)
```

get(int idx)

Do dowolnego elementu tablicy można się też łatwo odwołać za pomocą metody *get(int idx)*; poniższy kod źródłowy wyświetla po kolei zawartość wszystkich elementów listy.

```
for (int i = 0; i < v.size(); i++) System.out.println(v.get(i));
```

Należy zwrócić uwagę na to, że w przeciwieństwie do tablic argument określający lokalizację elementu na liście przekazywany jest jako argument metodym, więc znajduje się wewnątrz nawiasów okrągłych, a nie kwadratowych jak w przypadku tablic.