

Algorytmy i struktury danych

Laboratorium nr 4. Lista jednokierunkowa

Zad. 1 Zapoznaj się z częściową implementacją listy jednokierunkowej *TForwardList<E>* przedstawioną na wykładzie. Klasa posiada zaimplementowany konstruktor, oraz metody *size*, *add*, *remove* i *set*. Przy wykorzystaniu zaimplementowanych metod napisz kod:

- wypełniający listę naturalnymi liczbami parzystymi mniejszymi od 20. Następnie wyświetl na ekranie liczbę elementów przechowywanych w liście.
- Następnie wyzeruj liczby w węzłach o nieparzystym indeksie. Wyzerowane elementy powinny zostać wyświetlone w konsoli. Zakładamy że węzeł pierwszy ma indeks 0, drugi ma 1, trzeci ma 2 itp.
- Na końcu usuń wszystkie węzły i wyświetl rozmiar listy. Wyświetl w konsoli usuwane elementy.

Zad. 2 Zaimplementuj brakujące metody:

- boolean addFirst(E e)* – dodając kolejny element do listy na jej początek, metoda zwraca true jeżeli udało się poprawnie dodać element,
- void add(int index, E element)* – dodając element na pozycję index, węzeł który był wcześniej na pozycji index jest „przesuwany” na pozycję index+1. W razie niepoprawnego indeksu rzucający jest wyjątek,
- void clear()* – czyszcząca całą listę,
- E removeFirst()* – usuwająca i zwracająca pierwszy element listy, w razie braku elementów rzucający jest wyjątek,
- E removeLast()* – usuwająca i zwracająca ostatni element listy, w razie braku elementów rzucający jest wyjątek,
- E get(int index)* – zwracająca element znajdujący się w węźle o indeksie index
- int indexOf(E element)* – zwracająca index pierwszego napotkanego elementu pasującego do element.

Zad. 3 Rozszerz implementację klasy *TForwardList<E>* o metody:

- toString()* pozwalającą na wyświetlenie elementów przechowywanych w liście
- printAsArray()* pozwalającą na wyświetlenie elementów przechowywanych oraz ich indeksów, np. w postaci „list[0] = 5”

Zad. 4 Porównaj działanie listy *TForwardList* z implementacją listy dostępnej w bibliotece standardowej *java.util*. Wykorzystaj timer do sprawdzenia czasu działania list. Porównaj listy przy wczytywaniu danych z pliku. Do pomiaru szybkości wykonania kodu wykorzystaj metodę *System.currentTimeMillis()*

Przykładowo aby sprawdzić ile czasu zajmuje inkrementacja zmiennej w pętli należy:

```
long start = System.currentTimeMillis();
long a = 0;
for( int j = 0 ; j < 10000; j++){
    a += 1;
}
long stop = System.currentTimeMillis();
System.out.println( String.format(„Czas wykonania: %d ms”, stop - start ) );
```

W razie braku efektu zwiększ liczbę operacji.