

# 1 Lista dwukierunkowa

## 1.1 Dodawanie elementów (*insert*).

Start

Wczytaj *value* oraz *index*

**Jeżeli** nie ( $0 \leq index \leq size$ ) **to**  
| Wypisz błąd o niedozwolonym indeksie  
**koniec**

**w przeciwnym razie**

| *node*  $\leftarrow$  new *Node*(*value*)

| **Jeżeli** *size* = 0 **to**

| | *front*  $\leftarrow$  *node*  
| | *back*  $\leftarrow$  *node*

| **koniec**

| **w przeciwnym razie jeżeli** *index* = 0 **to**

| | *node.next*  $\leftarrow$  *front*  
| | *front.previous*  $\leftarrow$  *node*  
| | *front*  $\leftarrow$  *node*

| **koniec**

| **w przeciwnym razie jeżeli** *index* = *size* **to**

| | *node.previous*  $\leftarrow$  *back*  
| | *back.next*  $\leftarrow$  *node*  
| | *back*  $\leftarrow$  *node*

| **koniec**

| **w przeciwnym razie**

| | *current*  $\leftarrow$  *front*

| | **Dla** *i*  $\leftarrow$  0 **do** *index* - 1 **wykonuj**

| | | *current*  $\leftarrow$  *current.next*

| | **koniec**

| | *node.previous*  $\leftarrow$  *current*

| | *node.next*  $\leftarrow$  *current.next*

| | *current.next.previous*  $\leftarrow$  *node*

| | *current.next*  $\leftarrow$  *node*

| **koniec**

| *size*  $\leftarrow$  *size* + 1

**koniec**

Stop

## 1.2 Usuwanie elementów (*remove*).

```
Start
Wczytaj index
Jeżeli !( $0 \leq index < size$ ) to
| Wypisz błąd o niedozwolonym indeksie
koniec
w przeciwnym razie
| Jeżeli size = 1 to
| | node  $\leftarrow front$ 
| | front  $\leftarrow brak$ 
| | back  $\leftarrow brak$ 
| koniec
w przeciwnym razie jeżeli index = 0 to
| | node  $\leftarrow front$ 
| | node.next.previous  $\leftarrow brak$ 
| | front  $\leftarrow node.next$ 
| koniec
w przeciwnym razie jeżeli index = size - 1 to
| | node  $\leftarrow back$ 
| | node.previous.next  $\leftarrow brak$ 
| | back  $\leftarrow node.previous$ 
| koniec
w przeciwnym razie
| | current  $\leftarrow front$ 
| | Dla i  $\leftarrow 0$  do index - 1 wykonuj
| | | current  $\leftarrow current.next$ 
| | koniec
| | node  $\leftarrow current.next$ 
| | node.next.previous  $\leftarrow current$ 
| | current.next  $\leftarrow node.next$ 
| koniec
| size  $\leftarrow size - 1$ 
| Wypisz node.value
koniec
Stop
```

### 1.3 Wyszukiwanie (*index*).

```
Start
Wczytaj value
current  $\leftarrow$  front
Dla i  $\leftarrow$  0 do size wykonuj
    Jeżeli current.value = value to
        | Wypisz i
        | Stop
    koniec
    current  $\leftarrow$  current.next
koniec
Wypisz -1
Stop
```