

# Podstawy Programowania Komputerów

Lista cykliczna

14 grudnia 2018

Lista cykliczna to lista, w której ostatni element wskazuje na pierwszy element listy. Przykład listy cyklicznej przedstawia rys. 1. W liście cyklicznej  $n$ -elementowej dla  $i \in \{1, 2, \dots, n - 1\}$  element  $i$ -ty wskazuje na element  $(i + 1)$ -szy i ponadto element  $n$ -ty wskazuje na element pierwszy.

Zdefiniujmy typy dla zbudowania cyklicznej listy jednokierunkowej:

---

```
typedef int T;

/** element listy cyklicznej */
struct element
{
    /** wartość przechowywana w liście */
    T wartosc;
    /** wskaźnik na następny element listy */
    element * pNext;
};
```

---

W pustej liście wskaźnik `pHead == nullptr`. W liście jednoelementowej jedyny element listy wskaźnikiem `pNext` wskazuje na samego siebie.

## Zadania

Proszę zaimplementować następujące funkcje:

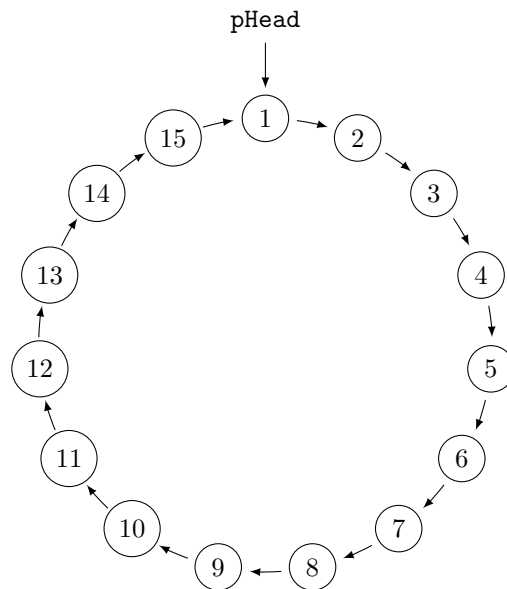
- ```
/** Funkcja dodaje element od cyklicznej listy jednokierunkowej na początek listy (przed element wskazywany przed pHead), co jest równocześnie dodaniem na koniec listy (po ostatnim elemencie).
@param pHead wskaźnik na pierwszy element listy
@param wartosc wartość do dodania do listy */
void dodaj (element * & pHead, const T & wartosc);
```

---
- ```
/** Funkcja usuwa wszystkie elementy z cyklicznej listy jednokierunkowej
@param pHead wskaźnik na pierwszy element listy */
void usun (element * & pHead);
```

---
- ```
/** Funkcja wypisuje na standardowe wyjście wartości elementów cyklicznej listy jednokierunkowej.
@param pHead wskaźnik na pierwszy element listy */
void wypisz (element * pHead);
```

---
- ```
/** Funkcja zlicza liczbę elementów cyklicznej listy jednokierunkowej.
@param pHead wskaźnik na pierwszy element listy
@return zwraca liczbę elementów listy, dla listy pustej zwraca zero */
int zliczElementy (element * pHead);
```

---



Rysunek 1: Przykład cyklicznej listy jednokierunkowej zawierającej 15 elementów.

- 
5. */\*\* Funkcja usuwa element o podanej wartości. Jeżeli w liście jest kilka takich elementów usuwa którykolwiek. \*/*  
*@param pHead wskaźnik na pierwszy element listy*  
*@param wartosc wartość elementu do usunięcia*  
*\*/*  
**void** usun (**element** \* & pHead, **const T** & wartosc);
- 
6. */\*\* Funkcja usuwa wszystkie elementy o podanej wartości. \*/*  
*@param pHead wskaźnik na pierwszy element listy*  
*@param wartosc wartość elementu do usunięcia*  
*\*/*  
**void** usunWszystkie (**element** \* & pHead, **const T** & wartosc);
- 
7. */\*\* Funkcja usuwa element o podanym adresie. \*/*  
*@param pHead wskaźnik na pierwszy element listy*  
*@param pDoUsuniecia adres elementu do usunięcia*  
*\*/*  
**void** usun (**element** \* & pHead, **const element** \* pDoUsuniecia);
- 
8. */\*\* Funkcja łączy dwie cykliczne listy jednokierunkowe. Po połączeniu list ostatni element pierwszej listy wskazuje na pierwszy element drugiej listy. Ostatni zaś element drugiej listy wskazuje na pierwszy element listy pierwszej. \*/*  
*@param pH1 wskaźnik na pierwszy element listy pierwszej, po wykonaniu funkcji wskaźnik wskazuje na listę wynikową*  
*@param pH2 wskaźnik na pierwszy element listy drugiej — dołączanej do pierwszej, po wykonaniu funkcji: nullptr*  
*\*/*  
**void** polacz (**element** \* & pH1, **element** \* & pH2);
-