

Zadanie 1

W programie \LaTeX , używanym powszechnie do składania tekstów, szczególnie naukowych i technicznych, instrukcje (komendy) mają zwykle postać $\text{\backslash command\{argument\}}$, gdzie argument może składać się z wielu słów. W szczególności $\text{\backslash index\{words\}}$ powoduje umieszczenie words jako hasła w indeksie na końcu książki czy artykułu.

Napisz program wczytujący plik \LaTeX 'owy i znajdujący w nim, za pomocą wyrażeń regularnych, wszystkie hasła umieszczone w komendach $\text{\backslash index\{...\}}$ i zapisujący znalezione wyrażenia do pliku tekstowego, każde hasło w osobnej linii. Zwróć uwagę, że hasło może się składać z wielu słów i zawierać znaki niealfanumeryczne. Może też być tak, że jeden fragment hasła wypada na końcu linii, a jego kontynuacja znajduje się już w linii następnej.

Na przykład dla pliku wejściowego zawierającego polski tekst w kodowaniu UTF-8 (nie zmieniaj podziału tekstu na linie)

```
Za początek okresu hellenistycznego uznaje się śmierć
\emph{Stagiryty}.\index{Stagiryta} To okres powstania
do dziś żywych koncepcji etycznych: \emph{epikureizmu}
(Epikur,\index{Epikur} ur. 341, zm. 270 p.n.e.)
mówiącego, że sensem życia jest doznawanie przyjemności
(por. \emph{hedonizm}) i \emph{stoicyzmu} (IV wiek p.n.e.
- II n.e.) dającego wzór postawy mędrca wobec śmierci
(Zenon z Kition,\index{Zenon
z Kition} Chryzyp, Seneka, mistrz retoryki Cynceron
i Marek Aureliusz\index{Marek
Aureliusz}, filozof na tronie). Sceptycy starożytni (Pirron,
\index{Pirron} Sekstus Empiryk)\index{Sekstus
Empiryk} z pobudek etycznych (uspokojenie umysłu) podawali
w wątpliwość możliwość wszelkiej wiedzy i dali początek wszelkim
nurtom sceptycznym. Tomasz z Akwinu\index{Tomasz
z Akwinu} (ur. 1225, zm. 1274)
```

wynik powinien być

```
Stagiryta
Epikur
Zenon z Kition
Marek Aureliusz
Pirron
Sekstus Empiryk
Tomasz z Akwinu
```

Zadanie 2

Napisz klasę **SingLList**, obiekty której reprezentują pojedynczo wiązane listy liczb typu **int**. Wewnątrz klasy zdefiniuj prywatną statyczną klasę **Node** opisującą pojedynczy węzeł listy — węzeł zawiera dane **data** (**int**) i referencję **next** wskazującą na następny węzeł listy. Taka prywatna statyczna klasa jest widoczna tylko w zakresie klasy otaczającej (w tym przypadku **SingLList**); jej pola są tam również widoczne bezpośrednio — *getery* są więc niepotrzebne. Sama klasa **SingLList** zawiera tylko jedno prywatne pole **head** typu **Node**, które wskazuje na pierwszy węzeł listy (być może jest to **null**, jeśli lista jest pusta).

Klasa **SingLList** zawiera funkcje:

```
1    public boolean empty()
2    public void addFront(int d)
3    public void addBack(int d)
4    public static SingLList arrayToList(int[] arr)
5    public void removeOdd()
6    public boolean contains(int d)
7    public void showList()
8    public void clear()
```

gdzie

1. metoda **empty** odpowiada na pytanie czy lista jest pusta;
2. metoda **addFront** dodaje nowy węzeł z daną **d** na początek listy;
3. metoda **addBack** dodaje nowy węzeł z daną **d** jako ostatni węzeł listy;
4. statyczna funkcja **arrayToList** pobiera tablicę **int**ów, tworzy i zwraca listę zawierającą podane w tablicy liczby jako dane (zachowując kolejność!);
5. metoda **removeOdd** usuwa z listy wszystkie węzły listy, w których dana **data** jest liczbą nieparzystą; jeśli wszystkie węzły zawierają liczby nieparzyste, lista stanie się pusta. Metoda nie może tworzyć żadnych nowych węzłów;
6. metoda **contains** odpowiada na pytanie czy lista zawiera węzeł zawierający daną równą;
7. metoda **showList** drukuje informacje o wszystkich węzłach listy, w jednej linii, oddzielone spacjami;
8. metoda **clear** czyni listę pustą.

Następująca funkcja **main**:

```
public static void main(String[] args) {
    int[] arr = {1,2,3,4,5,6,7};
    SingLList list = SingLList.arrayToList(arr);
    list.showList();
    list.removeOdd();
    list.showList();
    list.addFront(1);
    list.addBack(8);
    list.showList();
}
```

[download SingLList.java](#)

```
        System.out.println("contains 3? "+list.contains(3));  
        System.out.println("contains 8? "+list.contains(8));  
        list.clear();  
        list.showList();  
    }
```

powinna wydrukować

```
1 2 3 4 5 6 7  
2 4 6  
1 2 4 6 8  
contains 3? false  
contains 8? true  
Empty list
```
