

**Algorytmy i struktury danych**  
**Laboratorium nr 5. Kolejka klasyczna**

Zad. 1 Zapoznaj się z implementacją kolejki klasycznej *TQueue<E>* przedstawioną na wykładzie. Klasa posiada zaimplementowany konstruktor oraz metody *get*, *top*, *put* oraz *isEmpty()*. Zauważ także, że możliwe jest iterowanie danych przechowywanych w kolejce np. przy wykorzystaniu pętli *foreach*.

Następnie:

a) Zaproponuj klasę *TaskManager*, która udostępnia funkcjonalność zarządzania zadaniami. Zadanie jest reprezentowane przez łańcuch znaków. Klasa pozwala na dodanie nowego zadania do kolejki (boolean *addTask(String)*), wyświetlania wszystkich zadań do realizacji jako listę numerowaną (*toString()*) oraz usuwania pierwszego zadania z kolejki do realizacji (*String deleteTask()*).

b) Napisz program umożliwiający korzystanie z managera zadań w pętli. Program wyświetla stan managera zadań a następnie udostępnia opcje do wyboru:

1. Dodaj zadanie
2. Usuń zadanie
3. Zakończ

Ostatnia opcja powoduje zapis wszystkich niezrealizowanych zadań do pliku.

Przykładowo program może wyświetlić:

```
#####  
Zadania do realizacji:  
1.eat  
2.sleep  
3.code  
4.repeat
```

Wybierz opcję:

1. Dodaj zadanie.
2. Usuń zadanie.
3. Zakończ.

A użytkownik po wybraniu 2 spowoduje usunięcie z kolejki pierwszego zadania. A po wybraniu 1 będzie proszony o podanie nazwy zadania:

1

Podaj nowe zadanie: get salaty

Zad. 2. Rozszerz klasę `TQueue<E>` o implementację metod:

- a) `int size()` - określającą liczbę elementów kolejki.
- b) `boolean single()` – zwracającą informację czy kolejka jest jednoelementowa
- c) `boolean switchHeadTail()` – zamieniającą miejscami głowę z ogonem
- d) `join( E[] )` – dołączającą elementy z tablicy do kolejki
- e) `join(TQueue<E> )` – dołączającą elementy z innej kolejki
- f) nadpisującą metodę `toString()`, która pozwala na wyświetlenie elementów kolejki w przyjazdy dla użytkownika sposób.

Zad. 3 W pliku o nazwie *data.txt* znajdują się mieszane dane. Napisz program, który wyświetli na ekranie wszystkie cyfry znajdujące się w pliku w odwrotnej kolejności niż były zapisane w nim, a następnie wyświetli wszystkie litery w takiej samej kolejności w jakiej wystąpiły w pliku. Program powinien wyświetlić kolejne cyfry w odwrotnej kolejności a następnie kolejne litery po przecinku. Program może przeczytać dane z pliku tylko raz. Zastosuj odpowiednie struktury danych aby rozwiązać problem.

Zad 4. W pliku *opendatasites91819.csv* zapisane są dane o stronach internetowych z otwartym dostępem do danych gov w krajach/regionach<sup>1</sup>. Plik można przeczytać tylko raz. Następnie kolejno wyświetl na ekranie 1) i 2):

- 1) Informację o nazwie krajów (rekordy z International Country) i odpowiadających im stron internetowych w odwrotnej kolejności niż występują w pliku,
- 2) Informacje o nazwie stanów USA (rekordy z US State) i adresie strony internetowej w kolejności występowania w pliku.

---

<sup>1</sup> Link: <https://www.data.gov/open-gov/>