

## **AiSD/Lab3** **(2020/21)**

### **Temat: Kolejki i stosy (implementacje tablicowe)**

1. Implementacje kolejki cyklicznej
2. Implementacje kolejki naturalnej (wszystkie elementy są przesuwane po pobraniu elementu)
3. Implementacje stosu ograniczonego
4. Implementacje stosu tonącego
5. Zastosowania kolejki i stosu

#### **Uwaga:**

Zmiana w stosunku do wcześniejszych zapowiedzi spowodowana zmianą kolejności wykładów. Listy będą tematem wykładu nr 3.

### **Zadania**

#### **Zadanie 1**

Zaimplementować kolejkę ograniczoną. Do przechowywania elementów wykorzystać tablicę. Próba wstawienia elementu do kolejki pełnej powinna być sygnalizowana odpowiednim przerwaniem (FullQueueException). Próba pobierania elementu z pustej kolejki jest sygnalizowana przerwaniem EmptyQueueException. Implementacja powinna imitować naturalną kolejkę np. przed okienkiem pocztowym: po podejściu do okienka pierwszej osoby z kolejki, pozostałe osoby przesuwają się do przodu.

#### **Zadanie 2**

Zaimplementuj za pomocą tablicy stos dynamicznie zmieniający swoją wielkość. Podwaja wielkość gdy zajętość osiągnie  $\frac{3}{4}$  wielkości i zmniejsza o połowę gdy zajętość spadnie do  $\frac{1}{4}$  wielkości.

#### **Zadanie 3**

Zaimplementować stos przy pomocy dwóch kolejek.

#### **Zadanie 4**

Napisać program obliczania wyrażeń arytmetycznych zapisanych w pliku tekstowym w notacji infiksowej (poszczególne elementy wyrażenia są oddzielone spacją). Wyrażenie może zawierać cztery podstawowe operatory oraz nawiasy. Program powinien zawierać metody:

- przeczytanie wyrażenia z pliku,
- konwersja z postaci infiksowej na postfiksową (wykorzystać stos i kolejkę)
- obliczenie wartości wyrażenia (wykorzystać stos)
- wyświetlenie wyrażenia (postaci infix i postfix)

Wersja lite zadania: Wszystkie priorytety działań są wyznaczone przez nawiasy.

#### **Uwaga:**

We wszystkich zadaniach zademonstrować działanie programu.