

Lista zadań nr 7: **Kolejki priorytetowe, drzewa binarne, ...**

**Zadanie 1.**

Zaimplementuj kolejkę priorytetową o organizacji stogowej do zadania priorytetowej obsługi kolejki klientów przez trzech urzędników (A, B, C). Każdy klient w kolejce ma do załatwienia sprawę, która wymaga określonego z góry czasu  $t_i$  (założyć, że czas ten jest przypisany klientowi).

Zakłada się, że:

- każdy klient trafia na koniec kolejki z czasem „losowym”, w „losowym” momencie,
- do obsługi (przez pierwszego wolnego urzędnika; decyduje tu kolejność od A do C) wybierany jest klient o najkrótszym czasie obsługi  $t_i$  (jeśli takich klientów jest wielu, obsługiwany jest klient, który czeka najdłużej).

Zaimplementuj stosowny algorytm oraz zilustruj jego działanie w odpowiedniej klasie testowej.

**Zadanie 2.**

Zaimplementuj algorytmy obsługi binarnego drzewa wyszukiwawczego BST:

- wyszukiwania minimalnego i maksymalnego elementu w drzewie,
- wyszukiwania następnika/poprzednika dla węzła o podanej wartości (jeśli taki węzeł istnieje oraz jeśli istnieje jego następnik/poprzednik),
- wstawienie elementu do drzewa (dołączenie nowego wierzchołka – jeśli jest to możliwe),
- usunięcie wierzchołka o podanej wartości (jeśli istnieje w drzewie),
- wyprowadzenie (wyświetlenie) elementów drzewa w porządkach: *in-order*, *pre-order*, *post-order*.

Rozważ możliwość realizacji rekurencyjnej i iteracyjnej (tam, gdzie to jest możliwe).

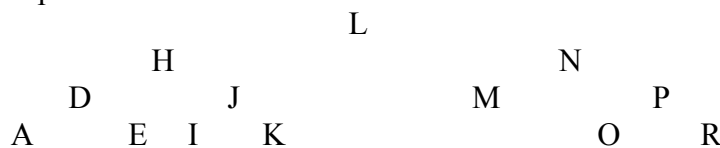
Zilustruj działanie poszczególnych metod w stosownej klasie testowej.

**Zadanie 3.**

Opracuj i zaimplementuj klasę realizującą wyświetlenie wierzchołków drzewa (zawierających liczby) w „naturalnej” postaci (bez krawędzi) – w kolejnych wierszach powinny znajdować się wierzchołki z poszczególnych poziomów drzewa (rozpoczynając od korzenia) a względny układ wierzchołków-przodków i ich potomków powinien być „czytelny”.

Należy przyjąć stosowne założenia dotyczące ograniczeń (w tym: maksymalna liczba wierzchołków) i zrealizować je.

Przykład postaci drzewa:



(w przykładzie zastosowano litery, ale należy użyć liczb całkowitych).

*Wskazówka: Można wykorzystać argument „format” metody printf, którego postać można budować na bieżąco, na podstawie postaci drzewa.*

Zilustruj efekty wykonania operacji z zadania 2. używając tej klasy.

**Zadanie 4.**

Zaimplementuj metodę wyznaczania współczynnika przeciążenia każdego wierzchołka drzewa (różnicę pomiędzy wysokościami prawego i lewego poddrzewa) i następnie, rozbudowując metodę z zadania 3., pokaż te współczynniki na „rysunku” wyprowadzanym przez program.