

Lista zadań nr 3: *Kolejki*

Zadanie 1.

Zaimplementuj ograniczoną kolejkę zwykłą (FIFO) z użyciem tablicy. Można w takiej realizacji uzyskać znaczną efektywność implementując kolejkę w sposób bezpośredni (bez wykorzystania klasy List). Zdefiniuj odpowiednią klasę np. ArrayFifoQueue. Złożoność wszystkich operacji powinna być stała, czyli $O(1)$.

Czym będzie różnić się od tej klasy klasa ArrayLifoQueue?

Zadanie 2.

Wykorzystując kolejkę zwykłą (FIFO) zasymuluj obsługę 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,
- klient z czoła kolejki trafia do tego urzędnika, który jest „wolny” (jeśli wielu jest wolnych, decyduje kolejność od A do C),

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

Zadanie 3. - dla hobbystów

Jest dany skład pociągu towarowego, jako ciąg połączonych ze sobą, kolejnych wagonów, stojących na bocznicach stacji węzłowej, w której rozpoczynają się dwie trasy (linie) kolejowe (o symbolach A i B). Każdy wagon jest opisany za pomocą unikatowego numeru, symbolu trasy oraz numeru stacji docelowej na tej trasie. Zakłada się, że numeracja kolejnych stacji na trasie jest uporządkowanym ciągiem kolejnych liczb naturalnych.

Z użyciem kolejek, symulujących bocznicę z rozjazdami (zwrotnicami) zrealizuj operację kompletowania dwóch składów pociągów (na trasę A i na trasę B) tak, by można było łatwo pozostawiać na stacjach docelowych stosowne wagony pociągów, odczepiając je z końca składów.

Przeanalizuj problem, przyjmij stosowne, dodatkowe założenia (ograniczenia).

Zaimplementuj stosowne klasy, w tym: klasę testową, która zilustruje działanie kompletowania składów.