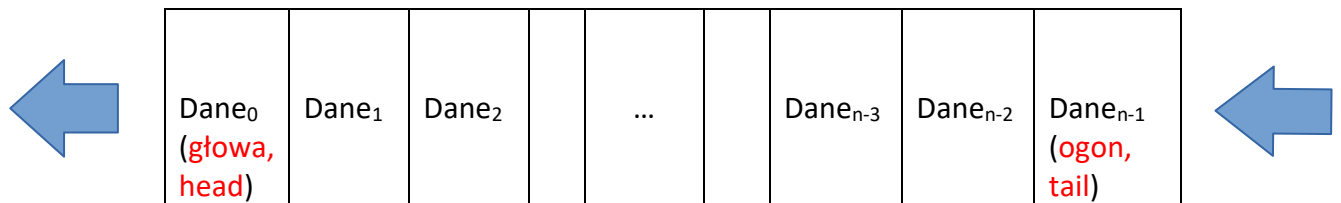


Kolejka – struktura języków programowania



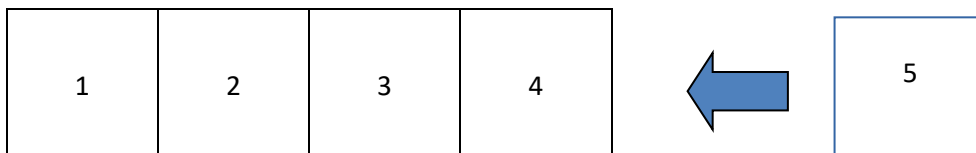
Opis kolejki: Kolejka to struktura danych, w której dane są poukładane liniowo, a dostęp do nich jest możliwy tylko z początku lub końca kolejki.

Kolejka to struktura danych typu FIFO (First-In, First-Out), gdzie pierwsze są pobierane dane, które do kolejki trafiły jako pierwsze.

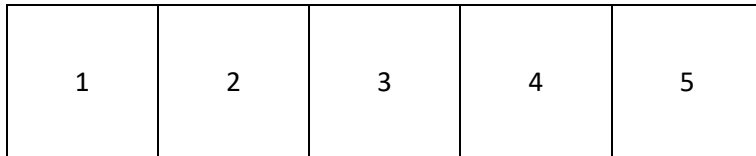


Konsekwencją tej reguły jest to, że elementy są dodawane do kolejki na tylko jednym z jej końców, a usuwane wyłącznie na drugim.

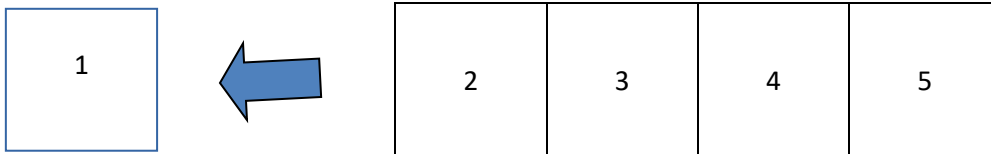
1. Dodanie elementu do kolejki



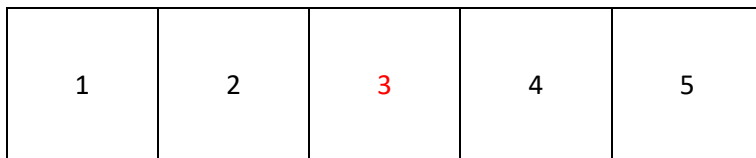
2. Kolejka jako zbiór danych



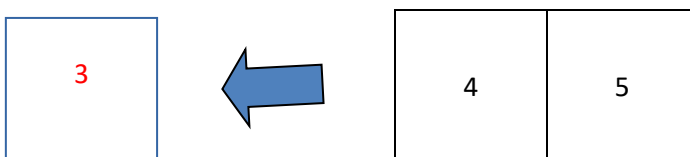
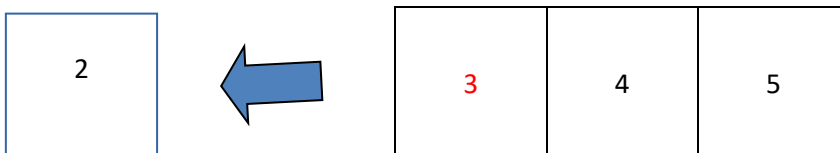
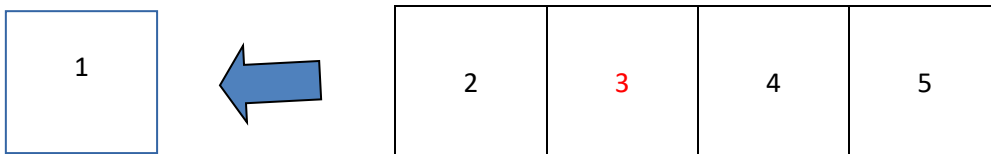
3. Pobranie elementu z kolejki



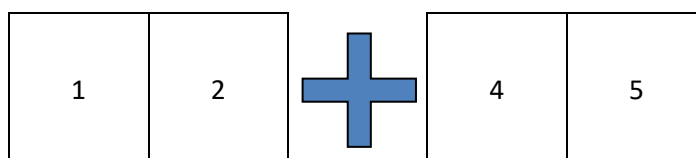
Chcąc dostać się do danych leżących w głębi kolejki, trzeba pobrać wszystkie dane znajdujące się przed nimi.



1. Pobieranie elementów z kolejki



W tym wypadku należy utworzyć nową kolejkę do której dodajemy elementy wcześniejsze. Po odczytaniu szukanych danych, kolejki należy skleić razem.



Zastosowanie:

Kolejkę spotyka się przede wszystkim w sytuacjach związanych z różnego rodzaju obsługą zdarzeń.

W systemach operacyjnych ma zastosowanie kolejka priorytetowa, przydzielająca zasoby sprzętowe uruchomionym procesom.

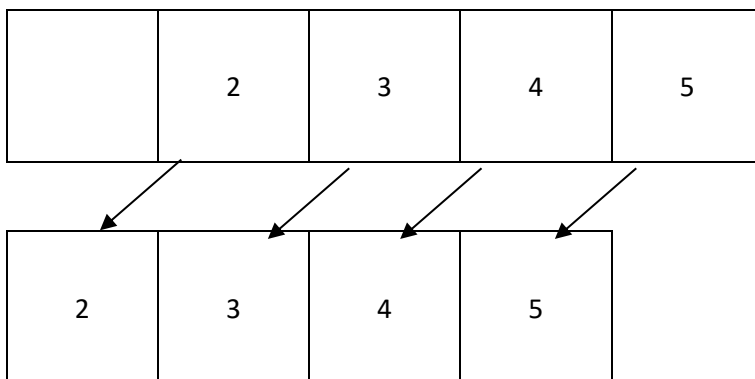
Operacje na kolejce:

- **Enqueue (obiekt)** (zakolejkuj) - czyli **dobanie obiektu na koniec kolejki**;
- **Dequeue ()** (odkolejkuj) - **pobranie pierwszego obiektu z początku kolejki** i zwrócenie jego wartości;
- **isEmpty()** – sprawdzenie czy kolejka jest pusta.
- **isFull()** – sprawdzenie czy kolejka nie jest przepełniona (czy ilość danych nie jest większa niż ilość miejsca na nie).

Implementacja kolejki:

1. Tablicowa.

- Tworzymy tablicę, która będzie kolejką.
- Dostęp do poszczególnych komórek będzie możliwy poprzez zmienną **licznik**, która jest indeksem ostatniego (najwyższego) elementu kolejki. Wskazuje na pierwszy pusty element.
- Pobieramy zawsze element z początku tablicy. Należy potem przesunąć wszystkie elementy od miejsca 1 do końca kolejki o jeden w lewo.



- gdy zmienna **licznik** wskazuje na komórkę 0 (początek kolejki), to próba pobrania nowego elementu wyświetli komunikat o pustej kolejce.
- gdy zmienna **licznik** wskazuje na komórkę poza kolejką, to próba dodania nowego elementu wyświetli komunikat o przepełnieniu kolejki.

2. Wskaźnikowa.

- Tworzymy tablicę, która będzie udostępniać miejsce dla kolejki. Nie musi ona obejmować całego zakresu tablicy!
- Tworzy się następnie wskaźniki, które będą wskazywały początek kolejki (**początek**), jej koniec (**koniec**), aktualnie wskazywaną pozycję (**licznik**).
- Dostęp do poszczególnych komórek będzie możliwy poprzez wskaźnik **licznik**, która jest indeksem ostatniego elementu kolejki. Wskazuje na pierwszy pusty element. Chcąc pobrać element skierować **licznik** na element wcześniejszy.
- Pobieramy zawsze element z początku tablicy. Należy potem przesunąć wszystkie elementy od miejsca 1 do końca kolejki, o jeden do przodu (zmniejszamy ich indeks o 1).
- gdy wskaźnik **licznik** wskazuje na tę samą komórkę co **początek**, to próba pobrania nowego elementu wyświetli komunikat o pustym stosie.
- gdy zmienna **licznik** skazuje na tę samą komórkę co **koniec**, to próba dodania nowego elementu wyświetli komunikat o przepełnieniu stosu.

3. Listowa.

- Tworzymy listę, która początkowo jest pusta.
- Korzystając z wbudowanych funkcji języków programowania, możemy dodawać na koniec elementy (polecenie **add**, **append** lub podobne) oraz usuwać z początku (polecenie **pop**, **remove** lub podobne z odpowiednim indeksem).
- Rozmiar możemy sprawdzić korzystając z wbudowanych poleceń (**size**, **len**, **length** lub podobne). W niektórych językach programowania lista może być nieskończona. W innych można zadeklarować jej maksymalny rozmiar.

Ćwiczenie:

1. Napisz program wczytujący łańcuch tekstowy i wprowadzający z niego znaki do kolejki.
2. Napisz funkcję zliczającą liczbę elementów umieszczonych w kolejce.

Zadanie

1. Napisz program, który wyszuka i pobierze z kolejki komórkę o numerze **x**. Pobranie nie może wiązać się z trwałym usunięciem komórek leżących wcześniej. Do ich tymczasowego przechowania użyj drugiej kolejki.
2. Napisz program, który odwróci kolejność elementów kolejki. Do ich odwrócenia użyj dodatkowej (lub dwóch) kolejki. Możesz też zastosować stos.
3. Napisz program, który wczyta zawartość dwóch kolejek o rozmiarach **n** i **m** do trzeciej (rozmiar **n + m**). Wczytywanie znaków ma się odbywać naprzemiennie.