

Kolejka – struktura języków programowania

M@я3k Pүđ€£kØ

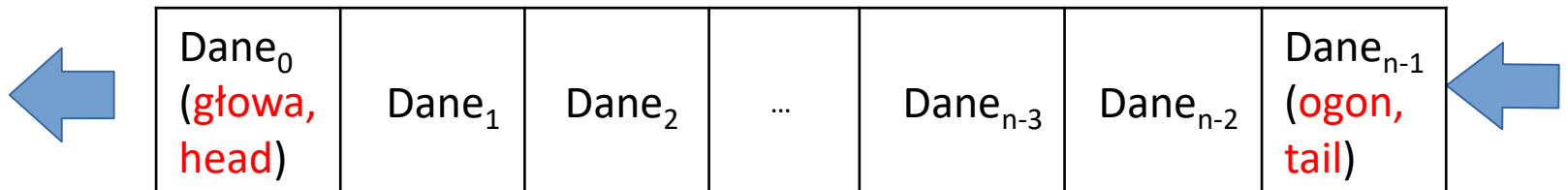
Programowanie w Pythonie

Spis treści

- Kolejka
- Zastosowanie kolejki
- Dodawanie/usuwanie elementów
- Operacje na kolejce
- Kolejka priorytetowa
- Implementacja listowa w pythonie

Kolejka

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

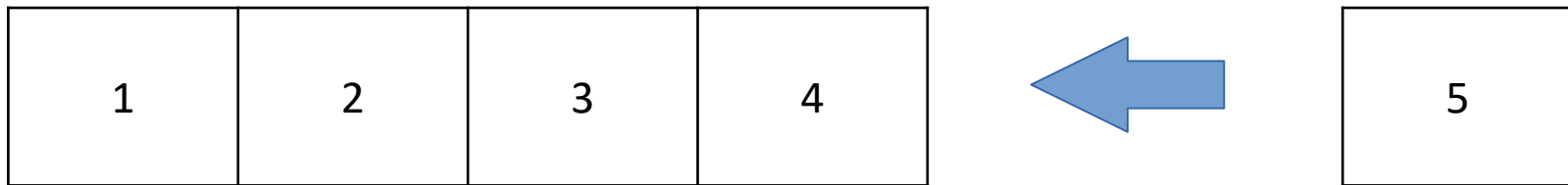


Zastosowanie kolejki

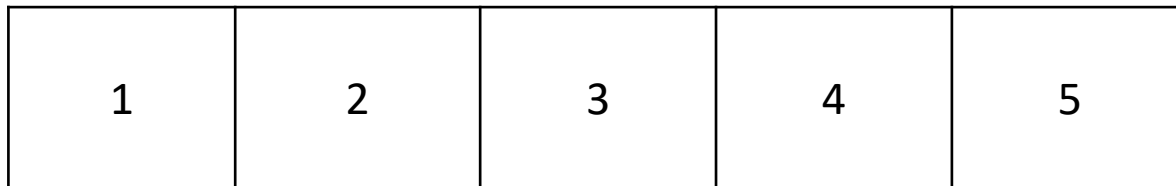
- **Zarządzanie zadaniami (Systemy operacyjne i aplikacje):**
 - Przechowywanie zadań czekających na wykonanie, np. w systemach kolejkowania zadań w tle (tzw. task queues).
- **Buforowanie (Buffering):**
 - Dane są generowane szybciej, niż mogą być przetwarzane (np. bufor klawiatury, drukowanie dokumentów, przesyłanie strumieniowe wideo).
- **Obsługa zdarzeń (Event Handling):**
 - Systemy graficzne (GUI) czy silniki gier używają kolejek do zarządzania zdarzeniami (kliknięcia myszką, naciśnięcia klawiszy), przetwarzając je w kolejności wystąpienia.
- **Algorytmy:**
 - Przeszukiwanie grafów i drzew w szerokość (Breadth-First Search - BFS) bazuje na kolejce do odwiedzania węzłów.
- **Komunikacja między procesami (IPC):**
 - W architekturze rozproszonej kolejki komunikatów (np. RabbitMQ, Kafka) umożliwiają bezpieczną wymianę danych między różnymi komponentami systemu.
- **Kolejka priorytetowa:**
 - Specjalny rodzaj kolejki, gdzie elementy z wyższym priorytetem są obsługiwane wcześniej, niezależnie od kolejności dodania.

Dodawanie/usuwanie elementów

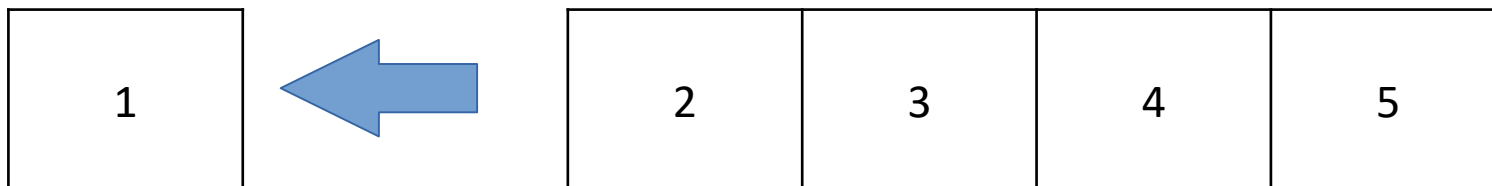
- Dodanie elementu



- Kolejka jako zbiór danych



- Pobranie elementu



Operacje na kolejce

- **Enqueue (obiekt)** (zakolejkuj) - czyli dodanie 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 przepelniona (czy ilość danych nie jest większa niż ilość miejsca na nie).

Kolejka priorytetowa

- Kolejka priorytetowa to struktura danych, która obsługuje elementy na podstawie ich ważności (priorytetu), a nie kolejności dodania. Najważniejsze elementy trafiają na początek kolejki, co umożliwia szybsze przetwarzanie zadań krytycznych.
 - Element o najwyższym priorytecie wychodzi pierwszy, niezależnie od czasu przybycia.
- **Implementacja:**
 - Często implementowana przy użyciu struktury kopca binarnego, co zapewnia efektywne dodawanie i usuwanie elementów.
- **Zastosowanie:**
 - Algorytmy grafowe (np. Dijkstra), planowanie zadań w systemach operacyjnych, przetwarzanie komunikatów o różnej ważności.
- W przypadku, gdy elementy mają identyczne priorytety, kolejka priorytetowa może zachowywać się jak FIFO dla tych konkretnych elementów.

Implementacja listowa w pythonie

- 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.

Przykład programu

```
kolejka = []
licznik = 0
rozmiar = 20
a = 1
print ("Zawartość kolejki", kolejka)
print ("Dodawanie do kolejki")
for i in range (rozmiar):
    kolejka.append (i*2)
    print (kolejka)
print ("Usuwanie z kolejki")
for i in range (rozmiar):
    a = kolejka.pop(0)
    print (kolejka)
```

Ćwiczenie 1

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

Ćwiczenie 2

1. Napisz program, który symuluje działanie kolejki w sklepie.
 - a) Utwórz kolejkę FIFO.
 - b) Dodaj do kolejki 5 klientów (np. imiona).
 - c) Następnie obsłuż dwóch pierwszych klientów (usuń ich z kolejki).
 - d) Wyświetl:
 - i. kto został obsłużony,
 - ii. kto jeszcze czeka w kolejce.
2. Napisz program symulujący łączenie dwóch (lub więcej) kolejek w jedną.
 - a) Łączenie klientów ma się odbywać naprzemiennie.
 - b) Kolejki mogą mieć różną długość.

Zadanie 1

1. Napisz program w Pythonie, który symuluje działanie kolejki w sklepie.
2. Program powinien posiadać menu, które działa w pętli aż użytkownik wybierze zakończenie programu.
3. Dostępne opcje:
 - a) Dodaj klienta do kolejki
 - b) Obsłuż klienta (usuń pierwszego z kolejki)
 - c) Pokaż pierwszy element kolejki
 - d) Wyświetl całą kolejkę
 - e) Sprawdź długość kolejki
 - f) Zakończ program
 - g) Dodatkowe warunki
4. Jeśli próbujemy obsłużyć klienta z pustej kolejki → program powinien wyświetlić komunikat.
5. Użyj struktury FIFO.
6. Wykorzystaj listę jako bazę.

Powtórzenie

- Co to jest kolejka?
- Jak wczytujemy dane do kolejki?
- Jak pobieramy dane z kolejki?
- Jak dostać się do danych wewnątrz kolejki?
- O jakich zabezpieczeniach należy pamiętać przy operacjach na kolejce?