



1
2
3
4

Stos – struktura języków programowania

M@я3k Pηđ€£kØ

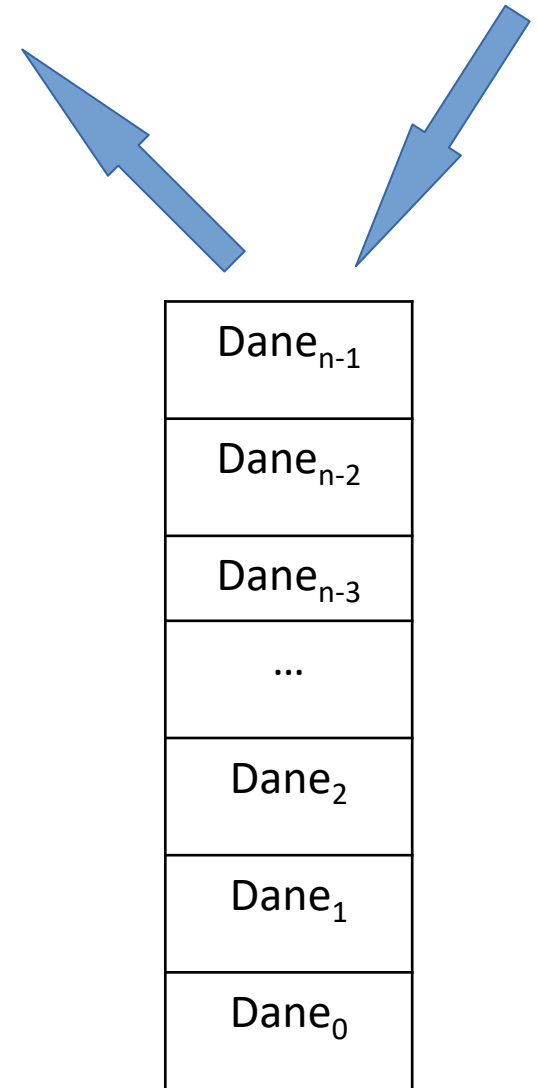
Programowanie w Pythonie

Spis treści

- Stos
- Zastosowanie stosu
- Dodawanie/usuwanie elementów
- Operacje na stosie
- Stos priorytetowy
- Implementacja listowa w pythonie

Stos

- Stos to struktura danych, w której dane są poukładane liniowo, a dostęp do nich jest możliwy tylko z wierzchołka stosu.
- Stos to struktura danych typu LIFO (Last-In, First-Out), gdzie pierwsze są pobierane dane, które trafiły jako ostatnie.
- Stos spotyka się przede wszystkim w sytuacjach związanych z różnego rodzaju obsługą zdarzeń.

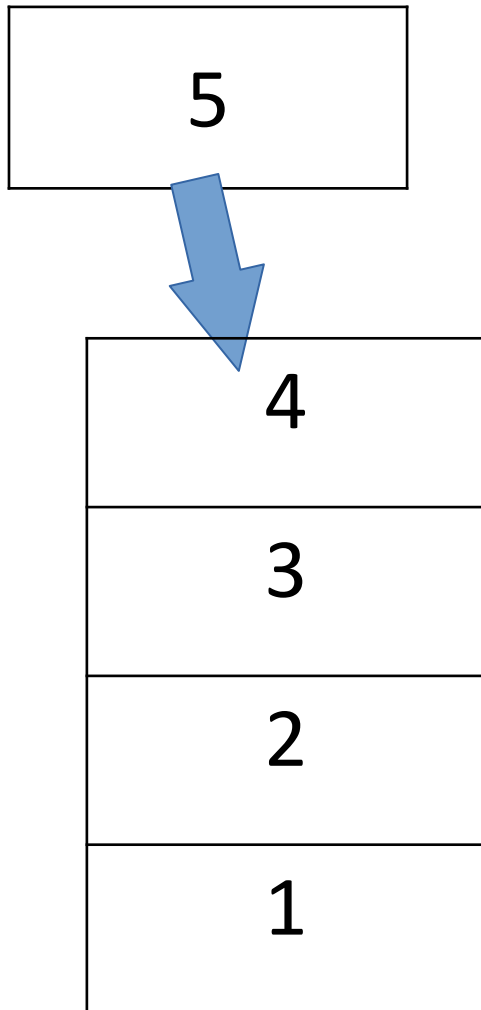


Zastosowanie Stosu

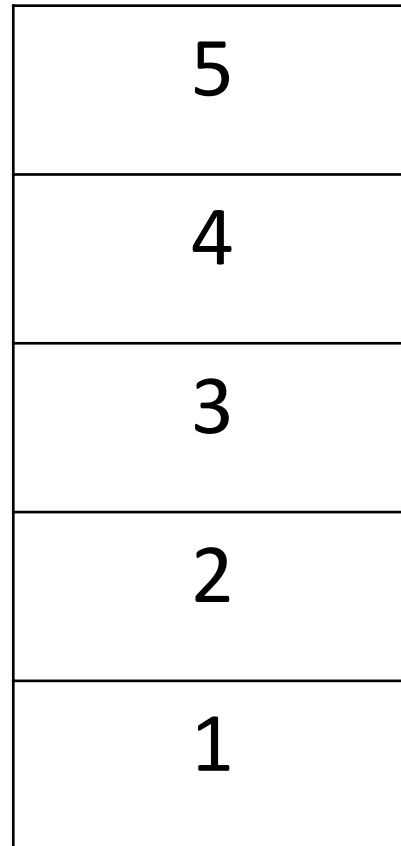
- **Zarządzanie wywołaniami funkcji (Call Stack):**
 - Gdy funkcja jest wywoływana, informacje o niej (argumenty, zmienne, adres powrotny) są odkładane na stos (push), a po zakończeniu – zdejmowane (pop). Najważniejsze zastosowanie w czasie wykonywania programu.
- **Algorytmy oparte na rekurencji:**
 - Stos to mechanizm umożliwiający działanie funkcji rekurencyjnych - przechowuje stany poszczególnych wywołań.
- **Algorytm odwrotnej notacji polskiej (ONP):**
 - Stos jest używany do konwersji wyrażeń infiksowych (np. $2 + 2$) na postać wykonalną przez komputer (np. $2\ 2\ +$) oraz do obliczania wartości takich wyrażeń.
- **Obsługa operacji "Cofnij" (Undo/Redo):**
 - W edytorach tekstu, programach graficznych stos przechowuje historię akcji. "Cofnij" zdejmuje ostatnią operację ze stosu.
- **Sprawdzanie poprawności nawiasów:**
 - W kompilatorach stos służy do sprawdzania, czy nawiasy ($()$, $[\]$, $\{ \}$) są poprawnie otwarte i zamknięte.
- **Przeszukiwanie grafów i drzew:**
 - Algorytmy takie jak DFS (ang. Depth-First Search – przeszukiwanie w głąb) wykorzystują stos do zapamiętywania ścieżki i węzłów do odwiedzenia.
- **Odwracanie danych:**
 - LIFO sprawia, że stos jest naturalnym narzędziem do odwracania kolejności elementów (np. odwrócenie napisu)
- **Stos priorytetowa:**
 - Specjalny rodzaj stosu, gdzie elementy z wyższym priorytetem są obsługiwane wcześniej, niezależnie od kolejności dodania.

Dodawanie/usuwanie elementów

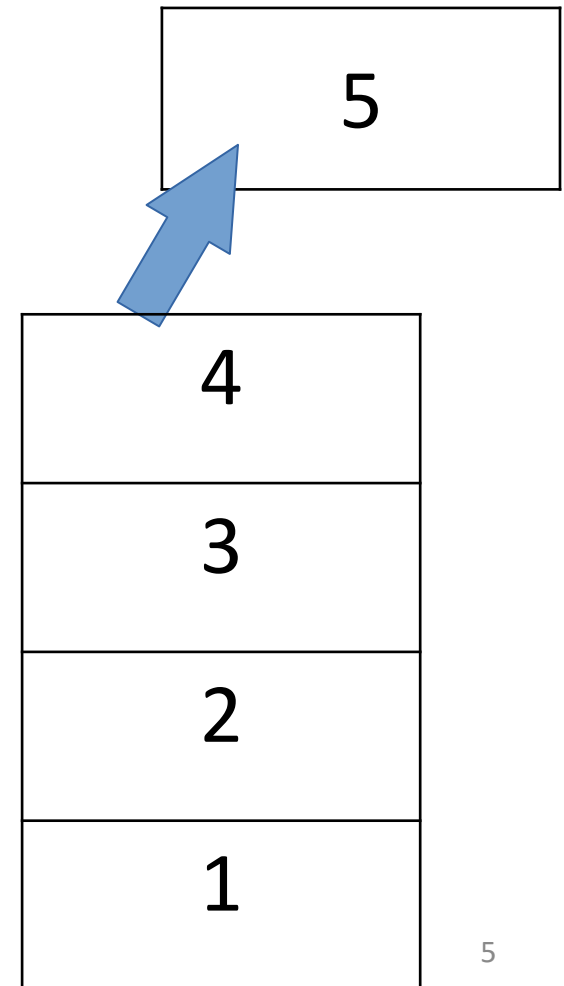
- Dodanie elementu



- Stos jako zbiór danych

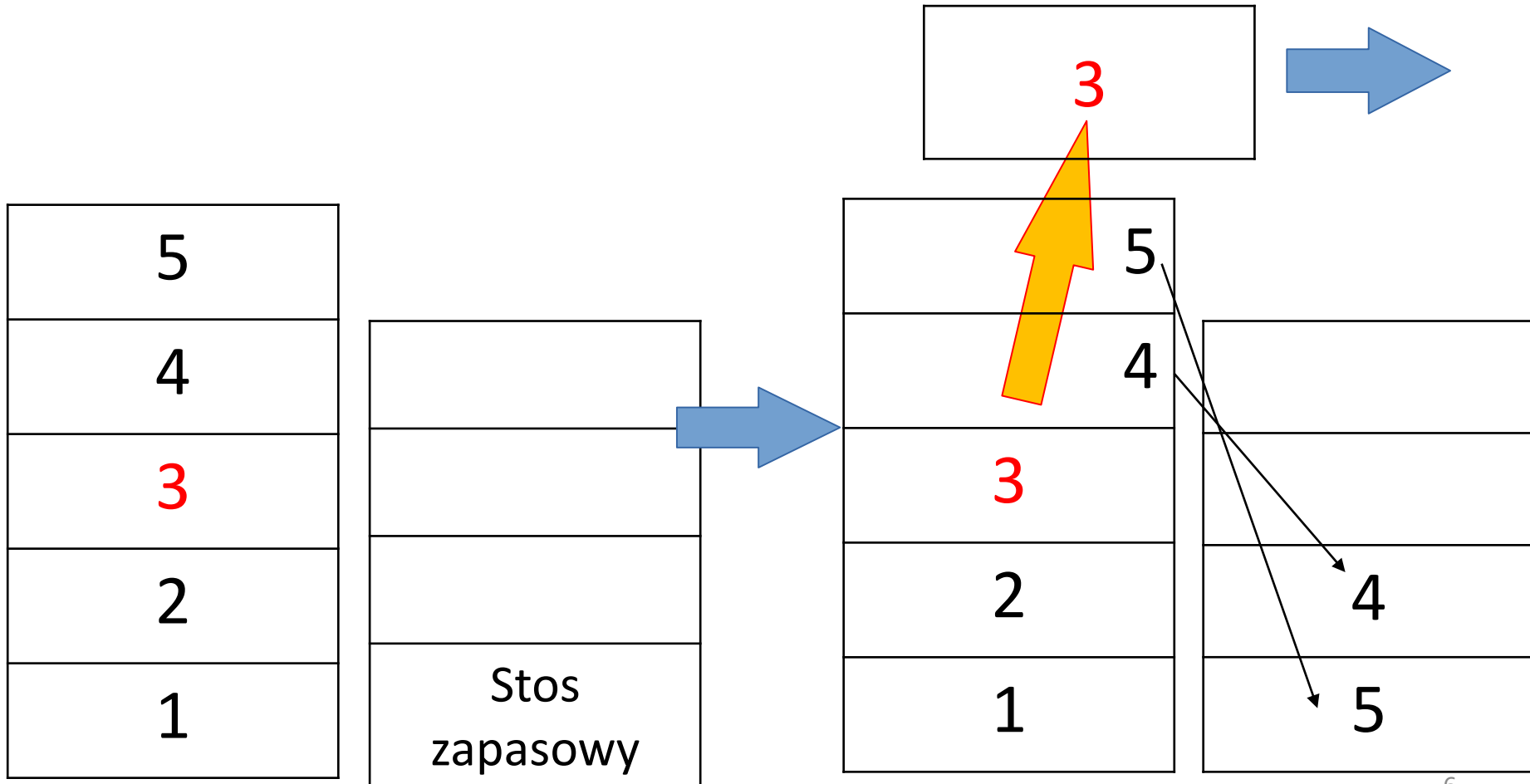


- Pobranie elementu



Dodawanie/usuwanie elementów leżących niżej

- Chcąc dostać się do elementu leżącego niżej trzeba ściągnąć ze stosu wszystkie dane leżące nad nimi.



Operacje na stosie

- **Push (obiekt)** - czyli dodanie obiektu na stos;
- **Pop ()** - pobranie pierwszego obiektu z wierzchu stosu i zwrócenie jego wartości;
- **isEmpty()** – sprawdzenie czy stos jest pusty.
- **isFull()** – sprawdzenie czy stos nie jest przepełniony (czy ilość danych nie jest większa niż ilość miejsca na nie).

Stos priorytetowy

- Stos priorytetowy to struktura danych, która działa podobnie do klasycznego stosu, ale każdy element ma przypisany priorytet.
- Element o wyższym priorytecie jest obsługiwany (usuwany/pobierany) przed elementem o niższym priorytecie, niezależnie od kolejności, w jakiej został dodany.
- **Kluczowe cechy i działanie**
 - Najważniejszy element jest zawsze na wierzchu i wychodzi pierwszy.
 - Najczęściej stosowaną strukturą do implementacji stosów/kolejek priorytetowych jest kopiec binarny (min-heap lub max-heap), co zapewnia efektywne operacje.
 - W przeciwieństwie do stosu (LIFO – ostatni wchodzi, pierwszy wychodzi) czy kolejki (FIFO – pierwszy wchodzi, pierwszy wychodzi), kolejność tutaj zależy od wartości priorytetu.
- **Zastosowania w programowaniu**
- Stosy i kolejki priorytetowe są kluczowe w algorytmach, gdzie szybkość reakcji na konkretne zadanie jest ważniejsza niż czas jego przyjścia:
 - **Szeregowanie zadań w systemach operacyjnych (RTOS):** Procesy o najwyższym priorytecie (np. obsługa przerw) są wykonywane jako pierwsze.
 - **Algorytmy grafowe:** Przykładem jest algorytm Dijkstry służący do znajdowania najkrótszej ścieżki, gdzie stos przechowuje wierzchołki z najmniejszym kosztem dojścia.
 - **Kolejki komunikatów:** Systemy zarządzające komunikatami, w których pilne wiadomości muszą wyprzedzić te mniej ważne.

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

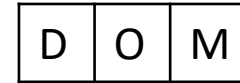
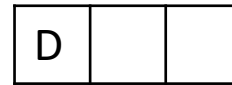
```
stos = []
licznik = 0
rozmiar = 20
a = 1
print ("Zawartość stosu", stos)
print ("Dodawanie do stos")
for i in range (rozmiar):
    stos.append (i*2)
    licznik += 1
    print (stos)
print ("Usuwanie ze stosu")
for i in range (rozmiar):
    a = stos.pop(licznik-1)
    licznik -= 1
    print (stos)
```

Ćwiczenie 1

1. Napisz program wczytujący łańcuch tekstowy i wprowadzający z niego znaki do stosu.
2. Napisz funkcję zliczającą liczbę elementów umieszczonych na stosie.
3. Napisz program, który wyszuka i pobierze ze stosu komórkę o numerze x .
 - a) Pobranie nie może wiązać się z trwałym usunięciem komórek leżących wyżej.
 - b) Do ich tymczasowego przechowania użyj drugiego stosu
4. Napisz program, który odwróci kolejność elementów leżących na stosie. Do ich odwrócenia użyj dodatkowego (lub dwóch) stosu. Możesz też zastosować tablicę
5. Napisz program, który wczyta zawartość dwóch stosów o rozmiarach n i m do trzeciego (rozmiar $n + m$).
Wczytywanie znaków ma się odbywać naprzemiennie.

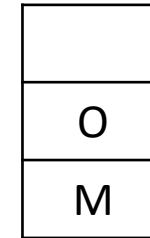
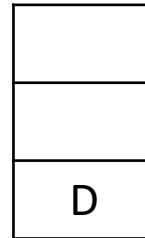
Zadanie 1

- W następującym ciągu litera oznacza operacją umieszczenia, a gwiazdka operacją pobrania elementu ze stosu



- Przykład**

D*MO** => DOM



- Napisz program czytający literał znaków i wykonujący według nich rozkazy. Podaj ciąg wartości zwracanych przez operację pobrania elementu ze stosu.

M*AJO***KZS***LO**A*

NOE*A*QYS***EU**IS*T****

ECO***A*ALTAN*****N*YT**C*IK**

Zadanie 2

- Anagramy
- W następującym ciągu litera oznacza operacją umieszczenia, a gwiazdka operacją pobrania elementu ze stosu:
 - umieść gwiazdki w ciągu ORAN tak, aby ciąg wartości zwracanych przez operacje pobrania elementu ze stosu był następujący: RANO.
 - MAKRELA -> REKLAMA
 - BRYKA -> RYBAK
 - ARBUZ -> BURZA

Zadanie 3

- Napisz program rozwiązujący działania matematyczne zapisane w odwrotnej notacji polskiej (kolejność **post-order**).
- Jeśli program napotka na liczby wczytuje je na stos.
- Pojawienie się działania oznacza, że pobiera 2 wcześniej położone liczby i wykonuje na nich działanie.
- Wynik zapisuje ponownie na stosie.
- **Przykład:**

wyrażenie: $a - b + (d + e) * c$ w porządku *post-order*
będzie mieć postać: $ab - de + c * +$

Powtórzenie

- Co to jest stos?
- Jak wczytujemy dane do stosu?
- Jak pobieramy dane ze stosu?
- Jak dostać się do danych wewnątrz stosu?
- O jakich zabezpieczeniach należy pamiętać przy operacjach na stosie?