

## Lekcja 24 - Stos - struktura języka C++

**Temat:** Omówienie stosu jako szczególnej struktury danych w językach programowania.

**Kod programu źródłowego:**

```
/******  
*****  
Stos - tablica  
*****  
*****/  
#include <iostream>  
#include <time.h>  
using namespace std;  
  
void wyswietl (int *stos, int licznik, int rozmiar)  
{  
    int i;  
    for (i=0; i < rozmiar; i++)  
    {  
        cout << stos[i] << " ";  
    }  
    cout << endl;  
}  
  
bool isEmpty (int licznik)  
{  
    //licznik wskazuje na pierwsze puste pole  
    if (licznik == 0)  
        return true;  
    else  
        return false;  
}  
  
bool isFull (int licznik, int rozmiar)  
{  
    //gdy wskazuje poza stos (indeksy stosu od 0 do n-1)  
    if (licznik == rozmiar) return true;  
    else  
        return false;  
}  
  
    //Przy wywołaniu funkcji przesyła się wskaźnik do tablicy,  
    //chcąc modyfikować zmienną licznik podaje się jej adres w pamięci  
void push (int *stos, int a, int &licznik, int rozmiar)  
{  
    if (isFull (licznik, rozmiar) == false)  
    {  
        stos[licznik] = a;  
        licznik++;  
    }  
}
```

```

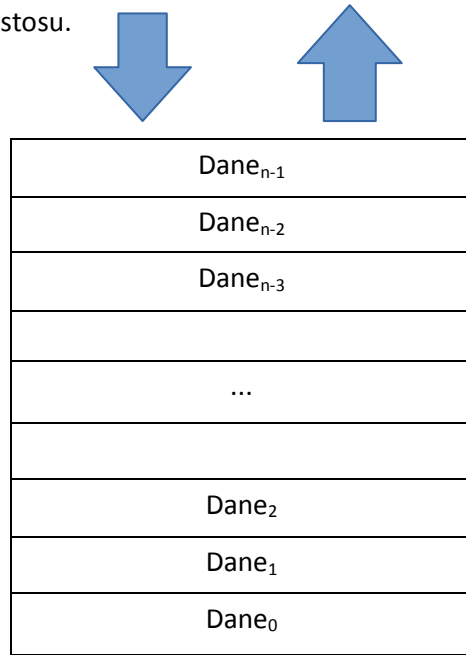
else
{
    cout <<"Błąd przepełnienia stosu \n";
}
}
//Przy wywołaniu funkcji przesyła się wskaźnik do tablicy,
//chcąc modyfikować zmienne a i licznik podaje się ich adresy w pamięci
void pop (int *stos, int &a, int &licznik)
{
    if (isEmpty (licznik) == false)
    {
        licznik--;
        a = stos[licznik];
        stos[licznik] = 0;
    }
    else
    {
        cout <<"Błąd! Stos jest pusty \n";
    }
}

int main()
{
    srand (time (NULL));
    int a, i, temp, rozmiar = 20, licznik = 0;
    int stos[rozmiar];

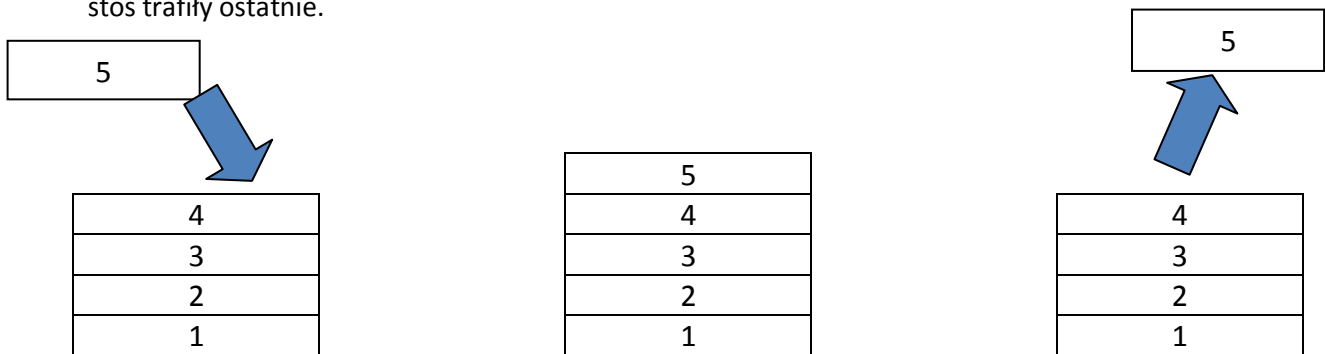
    for (i=0; i < 20; i++)
    {
        stos[i] = 0;
    }
    for (i=0; i < 10; i++)
    {
        a = (rand () % 100) + 1;
        push (stos, a, licznik, rozmiar);
    }
    wyswietl (stos, licznik, rozmiar);
    for (i=0; i < 5; i++)
    {
        pop (stos, a, licznik);
    }
    wyswietl (stos, licznik, rozmiar);
    return 0;
}

```

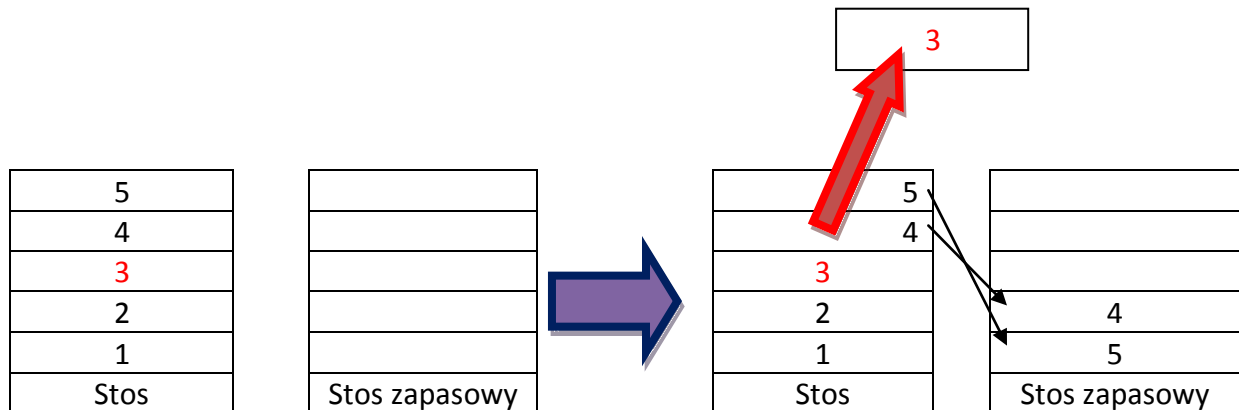
**Opis stosu:** Stos to struktura danych, w której dane są poukładane liniowo, a dostęp do nich jest możliwy tylko z wierzchołka stosu.



Stos to struktura danych typu LIFO (Last-In, First-Out), gdzie pierwsze są pobierane dane, które na stos trafiły ostatnie.



Chcąc dostać się do danych leżących niżej, trzeba ściągnąć ze stosu wszystkie dane leżące nad nimi.



## Zastosowanie:

Stos jest używany w systemach komputerowych na wszystkich poziomach funkcjonowania systemów informatycznych. Stosowany jest przez procesory do chwilowego zapamiętywania rejestrów procesora, do przechowywania zmiennych lokalnych, a także w programowaniu wysokopoziomym (np. Odwołania rekurencyjne).

## Operacje na stosie:

- `push(obiekt)` – czyli **odłożenie obiektu na stos**;
- `pop()` – **ściągnięcie obiektu ze stosu** i zwrócenie jego wartości;
- `isEmpty()` – sprawdzenie czy na stosie znajdują się już jakieś obiekty.
- `isFull()` – sprawdzenie czy stos nie jest przepelniony (czy ilość danych nie jest większa niż ilość miejsca na nie).

## Implementacja stosu:

### 1. Tablicowa.

- Tworzymy tablicę, która będzie stosem.
- Dostęp do poszczególnych komórek będzie możliwy poprzez zmienną **licznik**, która jest indeksem ostatniego (najwyższego) elementu stosu. Wskazuje na pierwszy pusty element. Chcąc pobrać element należy zmniejszyć licznik o 1.
- gdy zmienna **licznik** wskazuje na komórkę 0, to próba pobrania nowego elementu wyświetli komunikat o pustym stosie.
- gdy zmienna **licznik** wskazuje na komórkę poza tablicą, to próba dodania nowego elementu wyświetli komunikat o przepełnieniu stosu.

### 1. Wskaźnikowa.

- Tworzymy tablicę, która będzie udostępniać miejsce dla stosu. Stos nie musi obejmować całego zakresu tablicy!
- Tworzy się następnie wskaźniki, które będą wskazywały początek stosu (**początek**), jego koniec (**koniec**), aktualnie wskazywaną pozycję (**licznik**).
- Dostęp do poszczególnych komórek będzie możliwy poprzez wskaźnik **licznik**, która jest indeksem ostatniego (najwyższego) elementu stosu. Wskazuje na pierwszy pusty element. Chcąc pobrać element należy zmniejszyć **licznik** 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.

**Ćwiczenie:**

1. Napisz program wczytujący łańcuch tekstowy i wprowadzający z niego znaki do stosu.
2. Napisz funkcję zliczającą liczbę elementów umieszczonych na stosie.