

**WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI**
POLITECHNIKI RZESZOWSKIEJ

Michał Kruczek

Sprawozdanie z projektu nr 3

Spis treści

1 Wstęp.....	1
2 Opis podstaw teoretycznych zagadnienia.....	1
3 Opis szczegółów implementacji problemu.....	2
3.1. Biblioteki.....	2
3.2. Zmienne.....	2
3.3. Funkcje.....	2
4 Schematy blokowe oraz pseudokody poszczególnych operacji.....	3
4.1. Dodawanie elementów.....	3
4.2. Usuwanie elementów.....	4
4.3. Sprawdzanie czy stos jest pusty.....	4
4.4. Sprawdzanie czy stos jest pełny.....	5
4.5. Wyszukiwanie danego elementu.....	5
4.6. Zliczanie elementów.....	5
5 Podsumowanie.....	6

1 Wstęp

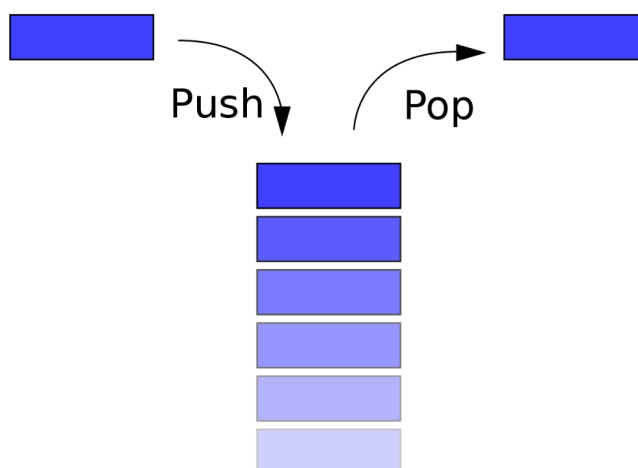
Treść zadania jest następująca:

Dokonaj implementacji struktury danych typu stos wraz z wszelkimi potrzebnymi operacjami charakterystycznymi dla tej struktury (inicjowanie struktury, dodawanie/usuwanie elementów, wyświetlanie elementów, zliczanie elementów/wyszukiwanie zadanego elementu).

2 Opis podstaw teoretycznych zagadnienia

Rozwiązanie zadania polegało na zaimplementowaniu struktury stosu oraz podstawowych operacji do wykonania na jego elementach. Do tego celu użyłem techniki programowania obiektowego oraz tablicy statycznej do przechowywania wartości występujących w stosie.

Sam stos jest natomiast linową strukturą danych, w której dane dokładane są na wierzch stosu i z wierzchołka stosu są pobierane. Nowy egzemplarz kładzie się na wierzch stosu i z wierzchu stosu zdejmują się kolejne egzemplarze. Elementy stosu poniżej wierzchołka można wyłącznie obejrzeć, aby je ściągnąć, trzeba najpierw po kolei ściągnąć to, co jest nad nimi.



3 Opis szczegółów implementacji problemu

3.1. Biblioteki

iostream	Deklaruje obiekty kontrolują odczytywanie ze strumieni standardowych i zapisywanie ich w tych strumieniach. Jest to często jedyny nagłówek potrzebny do wprowadzania danych i danych wyjściowych z programu w języku C++.
----------	---

3.2. Zmienne

wierzcholek	Przechowuje wierzchołek stosu
usuwana_wartosc	Przechowuje wartość wierzchołka podczas operacji usuwania
pozycja	Przechowuje indeks tablicy
wartosc	Przechowuje wartosc elementu w stosie
tab[6]	Tablica będąca podstawą stosu, ma określony z góry rozmiar
opcja	Służy do wyboru danej operacji w funkcji main
i	Licznik pętli for
ilosc	Zawiera ilosc kombinacji trzech wyrazów tablicy równych liczbie M

3.3. Funkcje

main	Funkcja programu, zawiera zbiór opcji pozwalających wykonać operacje określone w stosie
dodaj	Funkcja, która wykonuje operacje dodawania elementu do stosu
usun	Funkcja, która wykonuje operacje usuwania elementu ze stosu

Czy_jest_pelny	Funkcja, która sprawdza czy stos jest pełny
Czy_jest_pusty	Funkcja, która sprawdza czy stos jest pusty
zlicz	Funkcja, która wykonuje operacje zliczania elementów w stosie
wyszukaj	Funkcja, która wyświetla wskazany przez użytkownika element w stosie
wyświetl	Funkcja, która wyświetla elementy aktualnie znajdujące się w stosie

4 Schematy blokowe oraz pseudokody poszczególnych operacji

4.1. Dodawanie elementów

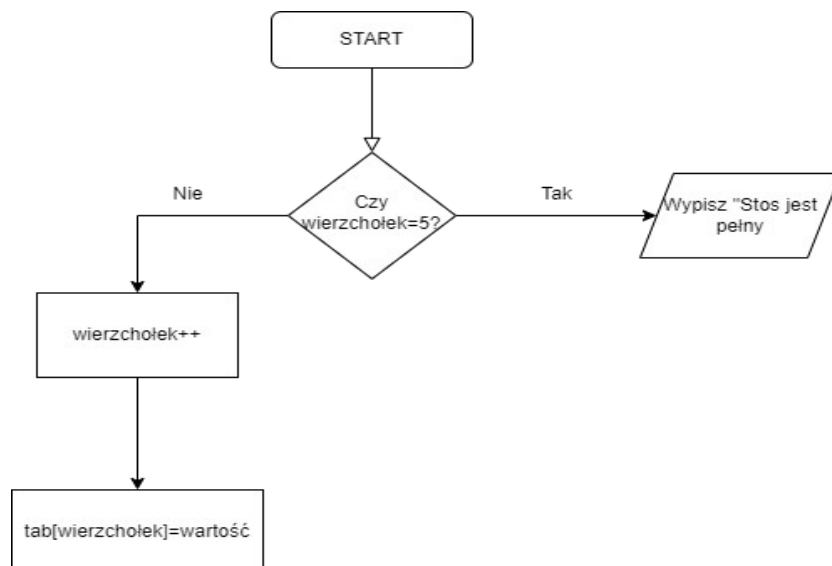
Jeśli wierzchołek = 5 to

wypisz „stos jest pełny”

w przeciwnym razie

wierzchołek++

tab[wierzchołek]=wartość



4.2. Usuwanie elementów

Jeśli wierzchołek = -1

wypisz "stos jest pusty"

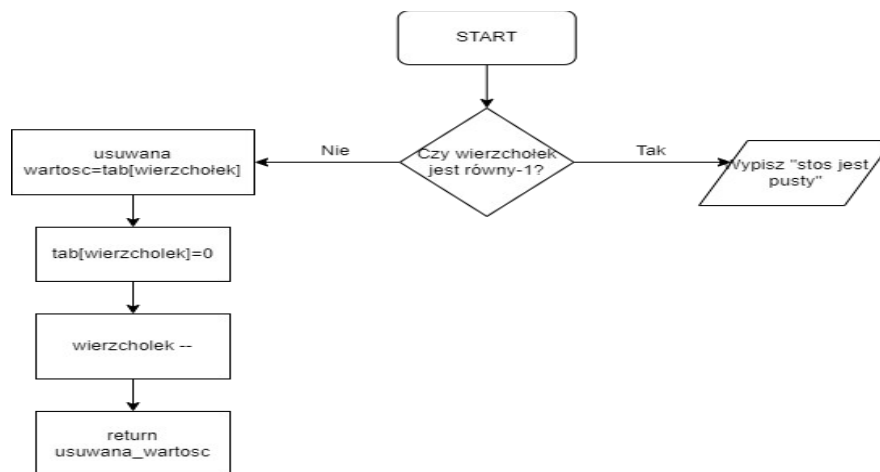
w przeciwnym razie

usuwana wartosc=tab[wierzchołek]

tab[wierzchołek]=0

wierzchołek--

return usuwana_wartosc



4.3. Sprawdzanie czy stos jest pusty

Jeśli wierzchołek = -1

return true

w przeciwnym razie

return false



4.4. Sprawdzanie czy stos jest pełny

Jeśli wierzchołek = 5

return true

w przeciwnym razie

return false



4.5. Wyszukiwanie danego elementu

Jeśli wierzchołek jest równy -1

wypisz „stos jest pusty”

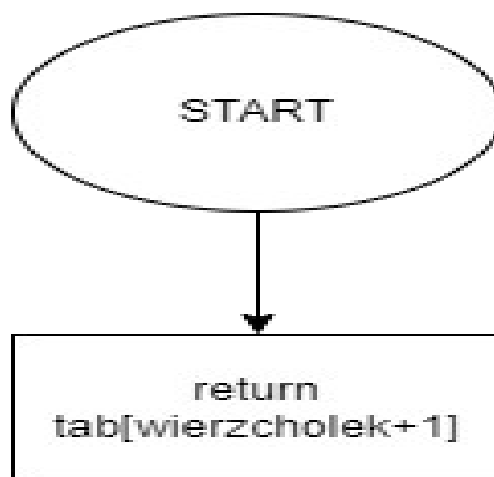
w przeciwnym razie

return tab[pozycja]



4.6. Zliczanie elementów

Return tab [wierzchołek+1]

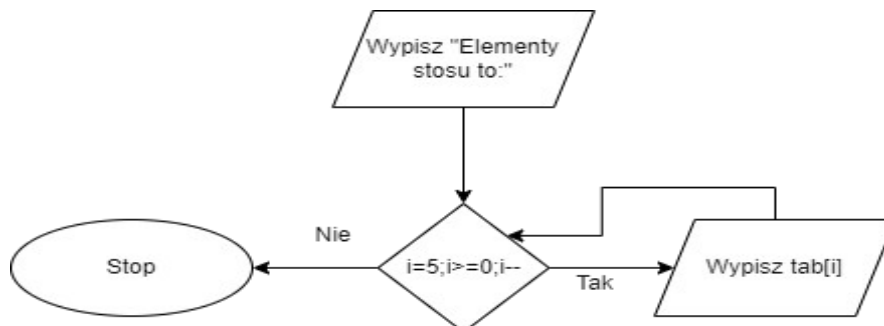


4.7. Wyświetlanie elementów

Wypisz „elementy stosu to”

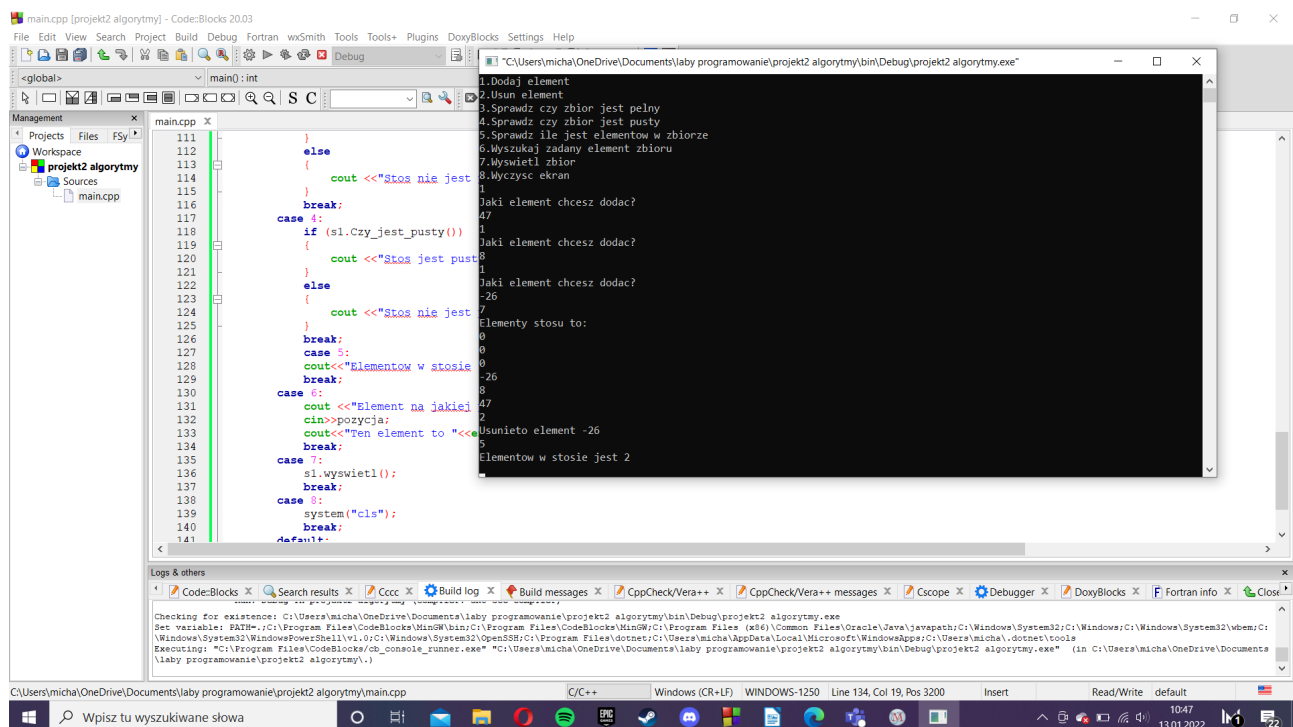
for i 5 to 0

wypisz tab[i]



5 Podsumowanie

Podsumowując, zaimplementowana przeze mnie struktura danych, spełnia wymagania stosu. Wykonany jest on za pomocą tablicy statycznej. Wykonuje on w sposób poprawny zaprogramowane operacje oraz za pomocą bardzo prostego interfejsu, pozwala użytkownikowi swobodnie modyfikować jego zawartość.



Ilustracja 2: Przykładowe uruchomienie programu