**SPRAWOZDANIE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Przedmiot** | Wprowadzenie do Informatyki | **Zadanie** | 5.2 |
| **Autor** | Andrii Godlevskyi | **Grupa** | WCY23KY1S1 |
| **Temat** | Struktury dynamiczne​ | | |

1.Treść zadania

Należy napisać program realizujący operacje   
**Stosu (LIFO)** zaimplementowanego wskaźnikowo :

* Top()– odczytuje element ze szczytu stosu
* Push(el)– odkłada element na szczyt stosu
* Pop()– zdejmuje element ze szczytu stosu
* IsEmpty()
* Clear()
* Listowanie elementów stosu

Metoda realizacji

Tworzymy stos i funkcje operujące na nim i w zależności od wyboru użytkownika wywołujemy potrzebną funkcje i wyprowadzamy wynik na ekran.

1. Założenia / ograniczenia dotyczące danych:

1. Dane wejściowe

Dane obiektów – wczytywane z klawiatury

1. Dane wyjściowe

Dane zapisanych obiektów - wyświetlone na ekranie *(z opisami).*

1. Realizacja
2. Algorytm

Изображение выглядит как текст, рукописный текст, бумага, Бумажное изделие

Автоматически созданное описание



1. Kod zródłowy

#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node\* next;

};

struct Stack {

struct Node \*top;

};

void initStack(struct Stack\* s) {

s->top = NULL;

}

void Pushel(struct Stack\* s, int value) {

struct Node\* temp = (struct Node\*)malloc(sizeof(struct Node));

temp->data = value;

temp->next = s->top;

s->top = temp;

}

void Pop(struct Stack\* s) {

if (s->top == NULL) {

printf("Stos jest pusty\n");

return;

}

struct Node\* temp = s->top;

s->top = s->top->next;

free(temp);

}

int IsEmpty(struct Stack\* s) {

return s->top == NULL;

}

void Clear(struct Stack\* s) {

while (!IsEmpty(s)) {

Pop(s);

}

}

void ListElements(struct Stack\* s) {

struct Node\* temp = s->top;

while (temp != NULL) {

printf("%d ", temp->data);

temp = temp->next;

}

printf("\n");

}

int Top(struct Stack\* s) {

if (s->top == NULL) {

printf("Stos jest pusty\n");

return -1;

}

return s->top->data;

}

int main() {

struct Stack s;

initStack(&s);

int choice, value;

do {

printf("\n1. Dodaj\n2. Pop\n3. Czy pusty?\n4. Oczysc\n5. Lista Elementow\n6. Top\n0. Wyjscie\n");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Wpisz wartosc do dodania: ");

scanf("%d", &value);

Pushel(&s, value);

break;

case 2:

Pop(&s);

break;

case 3:

if (IsEmpty(&s)) {

printf("Stos jest pusty\n");

} else {

printf("Stos nie jest pusty\n");

}

break;

case 4:

Clear(&s);

printf("Stos jest wyczysczony\n");

break;

case 5:

ListElements(&s);

break;

case 6:

printf("Top element: %d\n", Top(&s));

break;

case 0:

printf("Wyjscie z programu\n");

break;

default:

printf("Zly wybor,sprobuj ponownie\n");

}

} while (choice != 0);

return 0;

}

3. Dane wejściowe /wyjściowe

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

1. Złożoność obliczeniowa algorytmu

O(n)