ГУО «БГУИР»

Факультет информационных технологий и управления

Кафедра систем управления

Отчет по

Лабораторной работе №2

Динамическая структура СТЕК

Подготовил:

Студент гр.222401

Бык А.А.

Проверила:

Семижон Е.А.

Минск 2023

Цель: изучить алгоритмы работы с динамическими структурами данных в виде стека.

Вариант №3.

Написать программу по созданию, добавлению, просмотру и решению приведенных далее задач (в рассмотренных примерах это действие отсутствует) для однонаправленного линейного списка типа Stack. Реализовать сортировку стека методами, рассмотренными в подразделе 3.1.

Удалить из созданного списка отрицательные элементы.

Код:

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#include<iostream>

using namespace std;

#define \_CRT\_SECURE\_NO\_WARNING

struct Stack {

int info;

Stack\* next;

}\*first, \*t;

Stack\* InStack(Stack\*, int);

void View(Stack\*);

void removeNegativeElements(Stack\*\*);

void DellAll(Stack\*\*);

void DeleteNegativeElements(Stack\*\*);

int main() {

int choice = 0;

int amountOfElements = 0;

int i;

srand(time(NULL));

int element;

while (true)

{

printf("\n 1 - Create 2 - Add 3 - View 4 - Remove negative elements 5 - Clear memory 0 - Exit\n");

scanf\_s("%d", &choice);

switch (choice)

{

case 1: case 2:

if (choice == 1 && first != NULL)

{

printf("Clear memory!");

break;

}

printf("Enter amount of elements: ");

while (!(cin >> amountOfElements))

{

printf("---error---");

return 0;

}

//scanf\_s("%d", &amountOfElements);

for (i = 1; i <= amountOfElements; i++)

{

element = rand() % 100 - 50;

first = InStack(first, element);

}

if (choice == 1)

{

printf("Elements were created: %d", amountOfElements);

}

else printf("Elements were added: %d", amountOfElements);

break;

case 3:

if (!first)

{

printf("Stack is empty!");

break;

}

printf("-------------------------------------------- Stack --------------------------------------------");

View(first);

break;

case 4:

DeleteNegativeElements(&first);

printf("Negative numbers was deleted!");

break;

case 5:

if (first != NULL)

DellAll(&first);

printf("Memory cleared!");

break;

case 0:

if (first != NULL)

DellAll(&first);

printf("Program is over!");

return 0;

break;

}

}

return 0;

}

Stack\* InStack(Stack\* p, int element)

{

Stack \*t = new Stack;

t->info = element;

t->next = p;

return t;

}

void View(Stack\* p)

{

Stack \*t = p;

while (t != NULL)

{

printf("\n%d", t->info);

t = t->next;

}

}

void DellAll(Stack\*\* p)

{

while (\*p != NULL)

{

t = \*p;

\*p = (\*p)->next;

delete t;

}

}

void removeNegativeElements(Stack\*\* p)

{

while (\*p != NULL)

{

t = \*p;

\*p = (\*p)->next;

delete t;

}

}

void DeleteNegativeElements(Stack\*\* p)

{

Stack\* t;

while ((\*p)->info < 0)

{

t = \*p;

\*p = (\*p)->next;

delete t;

}

t = \*p;

while (t->next != NULL)

{

if (t->next->info < 0)

{

Stack\* I = t->next;

t->next = t->next->next;

delete I;

}

else t = t->next;

}

}

Вывод: изучила алгоритмы работы с динамическими структурами данных в виде стека.

