Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Реферат

По дисциплине «Основы алгоритмизации и программирования»

На тему «Реализация стека с помощью односвязного списка и динамического массива»

Выполнила:

Студент(ка) 1 курса 7 группы

Подшиваленко Диана Игоревна

Проверил:

Белодед Николай Иванович

2024, Минск

1. **Реализация стека с помощью динамического массива**

#include <iostream>

#include <string>

#include <Windows.h>

#include <regex>

using namespace std;

int\* stack = new int(100);

int top\_el = -1;

int stack\_size = 100;

regex checknum("[\\-]?[0-9]+");

void push(int);

void pop();

bool empty();

int top();

int size();

void resize(int);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char choice; // переменная для выбора опции

string c;

do {

cout << "Введите:\n";

cout << "1 - для добавления элемента в стек\n";

cout << "2 - для удаления элемента из стека\n";

cout << "3 - для проверки на пустоту\n";

cout << "4 - для вывода верхнего элемента\n";

cout << "5 - для вывода размера стека\n";

cout << "6 - для выхода\n";

getline(cin, c);

while (c == "") {

getline(cin, c);

}

while (c.length() != 1) {

cout << "Некорректный ввод\n";

getline(cin, c);

}

choice = c[0];

switch (choice) {

case '1': {

int n;

cout << "Введите число, которое хотите добавить в стек\n";

string b;

getline(cin, b);

while (!regex\_match(b, checknum)) {

cout << "Некорректный ввод\n";

getline(cin, b);

}

n = stoi(b);

push(n);

break;

}

case '2': {

pop();

break;

}

case '3': {

if (empty()) {

cout << "Стек пуст\n";

}

else {

cout << "Стек не пуст\n";

}

break;

}

case '4': {

int k = top();

if (k != -1) {

cout << k << '\n';

}

else {

cout << "Стек пуст\n";

}

break;

}

case '5': {

int k = size();

if (k == -1) {

cout << "Стек пуст\n";

}

else {

cout << "Количество элементов в стеке " << k << '\n';

}

break;

}

case '6':{

break;

}

default: {

cout << "Некорректный ввод\n";

break;

}

}

} while (choice != '6');

}

void push(int a) {

if (top\_el == stack\_size - 1) {

resize(stack\_size \* 2);

}

top\_el++;

stack[top\_el] = a;

}

void resize(int siz) {

int\* temp = new int(siz);

for (int i = 0; i < stack\_size; i++) {

temp[i] = stack[i];

}

delete[] stack;

stack = temp;

stack\_size = siz;

}

bool empty() {

return top\_el == -1;

}

void pop() {

if (empty()) {

return;

}

top\_el--;

}

int top() {

if (empty()) {

return -1;

}

return stack[top\_el];

}

int size() {

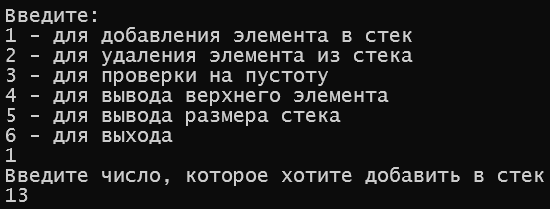
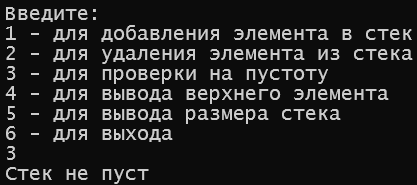
if (empty()) {

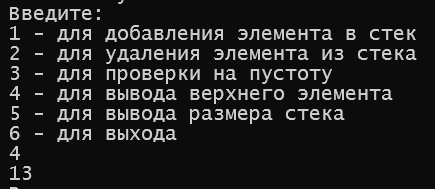
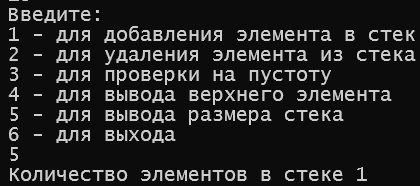
return -1;

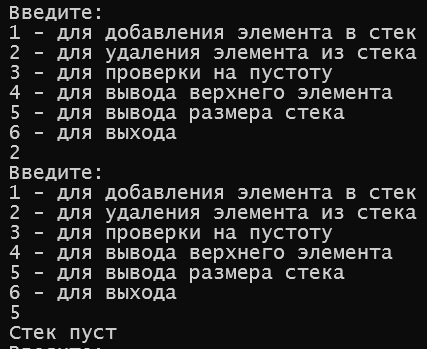
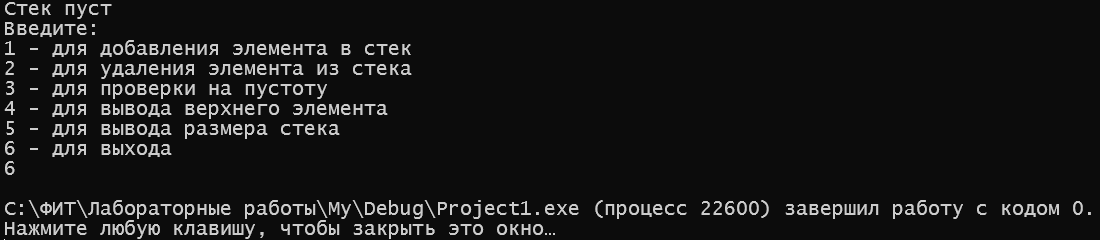
}

return top\_el + 1;

}

Результат выполнения:





1. **Реализация стека с помощью односвязного списка**

#include <iostream>

#include <string>

#include <Windows.h>

#include <regex>

using namespace std;

regex checknum("[\\-]?[0-9]+");

struct Mystack {

long long int num;

Mystack\* next = nullptr;

};

Mystack\* stack = nullptr;

void push(Mystack\*&, long long int);

void pop(Mystack\*&);

int top(Mystack\*&);

bool empty(Mystack\*&);

int size(Mystack\*);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char choice; // переменная для выбора опции

string c;

do {

cout << "Введите:\n";

cout << "1 - для добавления элемента в стек\n";

cout << "2 - для удаления элемента из стека\n";

cout << "3 - для проверки на пустоту\n";

cout << "4 - для вывода верхнего элемента\n";

cout << "5 - для вывода размера стека\n";

cout << "6 - для выхода\n";

getline(cin, c);

while (c == "") {

getline(cin, c);

}

while (c.length() != 1) {

cout << "Некорректный ввод\n";

getline(cin, c);

}

choice = c[0];

switch (choice) {

case '1': {

int n;

cout << "Введите число, которое хотите добавить в стек\n";

string b;

getline(cin, b);

while (!regex\_match(b, checknum)) {

cout << "Некорректный ввод\n";

getline(cin, b);

}

n = stoi(b);

push(stack,n);

break;

}

case '2': {

pop(stack);

break;

}

case '3': {

if (empty(stack)) {

cout << "Стек пуст\n";

}

else {

cout << "Стек не пуст\n";

}

break;

}

case '4': {

int k = top(stack);

if (k) {

cout << k << '\n';

}

break;

}

case '5': {

int k = size(stack);

cout << "Количество элементов в стеке " << k << '\n';

break;

}

case '6':{

break;

}

default: {

cout << "Некорректный ввод\n";

break;

}

}

} while (choice != '6');

}

void push(Mystack\*& st, long long int n) {

Mystack\* newst = new Mystack;

newst->num = n;

newst->next = st;

st = newst;

}

void pop(Mystack\*& st) {

if (st == NULL) {

cout << "Стек пуст\n";

return;

}

else {

Mystack\* newst = st;

st = st->next;

delete newst;

}

}

int top(Mystack\*& st) {

if (st == NULL) {

cout << "Стек пуст\n";

return 0;

}

else {

return st->num;

}

}

bool empty(Mystack\*& st) {

return (st == NULL);

}

int size(Mystack\* st) {

long long int s = 0;

while (st) {

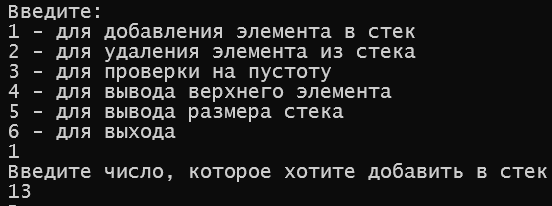
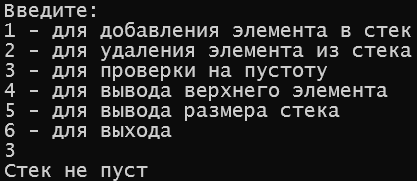
s++;

st = st->next;

}

return s;

}

Результат выполнения:

