**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Программирование»**

Тема: **Динамические структуры данных. Тестирование.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Рыбин А.С. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

Оглавление

[**Оглавление** 2](#_Toc479199556)

[Цель: 3](#_Toc479199557)

[Задание: 3](#_Toc479199558)

[Содержание: 3](#_Toc479199559)

[Инициализация 3](#_Toc479199560)

[Разбиение на лексемы 3](#_Toc479199561)

[Работа стековой машины 3](#_Toc479199562)

[**выводы** 5](#_Toc479199563)

[Приложение 1 6](#_Toc479199564)

[Приложение 2 7](#_Toc479199565)

[Приложение 3 8](#_Toc479199566)

Цель:

Написание программы с использованием динамических структур данных.

Задание:

**Стековая машина.**

На вход программе подается последовательность (не более 100 элементов) из чисел и арифметических операций (+, -, \*, / (деление нацело)) разделенных пробелом, которые программа должна интерпретировать и выполнить по следующим правилам:

* Если очередной элемент входной последовательности - число, то положить его в стек
* Если очередной элемент - знак операции, то применить эту операцию над двумя верхними элементами стека, а результат положить обратно в стек (следует считать, что левый операнд выражения лежит в стеке глубже)
* Если входная последовательность закончилась, то вывести результат (число в стеке)

Если в процессе вычисления возникает ошибка (для операции в стеке не хватает аргументов или по завершении работы программы в стеке более одного элемента), то вместо результата следует вывести "error"

Стек требуется реализовать самостоятельно на базе массива.

Содержание:

Инициализация

int stek[100];

/\* Index of last element in stek \*/

int end = -1;

char buffer[200];

fgets(buffer, 200, stdin);

Разбиение на лексемы

char\* ptr = strtok(buffer, " ");

while (ptr)

{

/\* Do integer from char \*/

int element = atoi(ptr);

/\* If integer value push \*/

if (!(element == 0))

push(stek, &end, element);

else

{

Работа стековой машины

/\* If too few arguments \*/

if (end < 1)

{

printf("error");

return 0;

}

int b = pop(stek, &end);

int a = pop(stek, &end);

switch (ptr[0])

{

case '+':

push(stek, &end, a + b);

break;

case'-':

push(stek, &end, a - b);

break;

case '\*':

push(stek, &end, a \* b);

break;

case '/':

push(stek, &end, a / b);

break;

default:

{

printf("error");

return 0;

}

}

}

ptr = strtok(NULL, " ");

}

int result = pop(stek, &end);

/\* Check if any elements exists in stek \*/

if (end > -1)

{

printf("error");

return 0;

}

else

printf("%d", result);

return 0;

}

# Вывод

В ходе выполнения лабораторной работы, используя стандартные средства языка С, была самостоятельно смоделирована такая динамическая структура данных как стек (в данном случае на базе массива), которая позволила удобно хранить и обрабатывать данные для решения поставленной задачи.

Приложение 1

/\*!

\file

\brief Lab2\_Sem2 Header

\author Rybin Aleksandr 1 course 2 half

\date 05.04.2017

\version 1.0

\*/

#include <stdlib.h>

#include <math.h>

#include <string.h>

#include <stdio.h>

#include <ctype.h>

/\*!

\brief Push element in stek

\param[in] stek Pointer to stek vector

\param[in] index Pointer to index of end of stek vector

\param[in] element Object to push

\return void

\ingroup Stek

\*/

void push(int\* stek, int\* index,int element);

/\*!

\brief Pop element from stek

\param[in] stek Pointer to stek vector

\param[in] index Pointer to index of end of stek vector

\return tmp Last object in stek

\ingroup Stek

\*/

int pop(int\* stek, int\* index);

Приложение 2

/\*!

\file

\brief Lab2\_Sem2 Functions for stek

\author Rybin Aleksandr 1 course 2 half

\date 05.04.2017

\version 1.0

\*/

#include "Rybin\_Lab2\_Sem2.h"

void push(int\* stek, int\* index,int element)

{

stek[\*index + 1] = element;

(\*index)++;

}

int pop(int\* stek, int\* index)

{

int tmp = stek[\*index];

(\*index)--;

return tmp;

}

Приложение 3

int main()

{

int stek[100];

/\* Index of last element in stek \*/

int end = -1;

char buffer[200];

fgets(buffer, 200, stdin);

char\* ptr = strtok(buffer, " ");

while (ptr)

{

/\* Do integer from char \*/

int element = atoi(ptr);

/\* If integer value push \*/

if (!(element == 0))

push(stek, &end, element);

else

{

/\* If too few arguments \*/

if (end < 1)

{

printf("error");

return 0;

}

int b = pop(stek, &end);

int a = pop(stek, &end);

switch (ptr[0])

{

case '+':

push(stek, &end, a + b);

break;

case'-':

push(stek, &end, a - b);

break;

case '\*':

push(stek, &end, a \* b);

break;

case '/':

push(stek, &end, a / b);

break;

default:

{

printf("error");

return 0;

}

}

}

ptr = strtok(NULL, " ");

}

int result = pop(stek, &end);

/\* Check if any elements exists in stek \*/

if (end > -1)

{

printf("error");

return 0;

}

else

printf("%d", result);

return 0;

}