**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.**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6304 |  | Иванкова В.М. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

Оглавление

[Цель: 3](#_Toc484110814)

[Задание: 3](#_Toc484110815)

[Содержание: 3](#_Toc484110816)

[Инициализация 3](#_Toc484110817)

[Разбиение на лексемы 3](#_Toc484110818)

[Работа стековой машины 3](#_Toc484110819)

[Вывод 5](#_Toc484110820)

[Приложение 6](#_Toc484110821)

Цель:

Написать программу, на вход которой подается последовательность из чисел и арифметических операций, разделенных пробелом, которые программа должна интерпретировать и выполнить по определенным правилам. Стек, который требуется реализовать самостоятельно, требуется выполнить на базе массива.

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 stack[100];

int end = -1;

char array[100];

fgets(array, 100, stdin);

Разбиение на лексемы

char\* str = strtok(array, " ");

while (str!=0)

{

int element = atoi(str);

if (element!= 0)

push(stack, &end, element);

else

{

Работа стековой машины

if (end < 1)

{

printf("error");

return 0;

}

int b = pop(stack, &end);

int a = pop(stack, &end);

switch (str[0])

{

case '+':

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

break;

case'-':

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

break;

case '\*':

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

break;

case '/':

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

break;

default:

{

printf("error");

return 0;

}

}

}

str = strtok(NULL, " ");

}

int result = pop(stack, &end);

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

#include <stdlib.h>

#include <math.h>

#include <string.h>

#include <stdio.h>

#include <ctype.h>

void push(int\* stack, int\* size,int element)

{

stack[\*size+1] = element;

(\*size)++;

}

int pop(int\* stack, int\* size)

{

int str = stack[\*size];

(\*size)--;

return str;

}

int main(){

int stack[100];

int end = -1;

char array[100];

fgets(array, 100, stdin);

char\* str = strtok(array, " ");

while (str!=0)

{

int element = atoi(str);

if (element!= 0)

push(stack, &end, element);

else

{

if (end < 1)

{

printf("error");

return 0;

}

int b = pop(stack, &end);

int a = pop(stack, &end);

switch (str[0])

{

case '+':

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

break;

case'-':

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

break;

case '\*':

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

break;

case '/':

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

break;

default:

{

printf("error");

return 0;

}

}

}

str = strtok(NULL, " ");

}

int result = pop(stack, &end);

if (end > -1)

{

printf("error");

return 0;

}

else

printf("%d", result);

return 0;

}