Министерство образования Республики Беларусь
Учреждение образования «Полоцкий государственный университет»

Кафедра технологий программирования

Алгоритмы и структуры данных

Отчет по лабораторной работе №2

Вариант 11

Выполнил — Ланцев Евгений Николаевич. 21-ИТ-1, ФИТ

Проверил — преподаватель Виноградова А.Д.

Полоцк
2022 г.

Лабораторная работа № 2
“Реализация линейной структуры данных «Стек» и основные алгоритмы обработки.”

Цель работы: ознакомиться с основами линейной структуры данных «Стек», изучить основные алгоритмы обработки ЛСД «Стек», научиться применять полученные знания на практике.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ (ответы на контрольные вопросы):

1. Определение понятия стек..

Стеком называется упорядоченный набор элементов, в котором размещение новых и удаление существующих происходит с одного конца, называемого вершиной.

2. Опишите принцип работы стека.

Принцип работы стека сравнивают со стопкой листов бумаги: чтобы взять второй сверху, нужно снять верхний.

3. Виды стека.

Аппаратный стек используется для хранения адресов возврата из функций и их аргументов.

Программный стек – это пользовательская модель (структура) данных.

4. Операции для работы со стеком.

- инициализация стека init(s), где s — стек
- помещение элемента в стек push(s, i), где s — стек, i — помещаемый элемент;
- удаление элемента из стека i=pop(s);
- получение верхнего элемента стека без его удаления i=stkTop(s),где s — стек
- получение количества элементов стека
- определение, пуст ли стек isempty(s) возвращает 1 если стек пустой и 0 в противном случае.
- вывод элементов стека stkPrint(s), где s — стек

5. Перечислите способы реализации стека.
    - с помощью одномерного массива;
    - с помощью связанного списка;
    - с помощью класса объектно-ориентированного программирования.

6. Для чего используется аппаратный стек?

Аппаратный стек используется для хранения адресов возврата из функций и их аргументов.

Вариант 11

```
const stack = [];
```

Рисунок 1 - Стек

```
if (task == 1) {
  rl.question("\nADD ELEMENT : ", (data) => {
    stack.push(data);
    tasks();
  });
} else if (task == 2) {
  console.log("\nDISPLAY : \n");
  console.log(stack);
  tasks();
} else if (task == 3) {
  x = 0;
  for (let i = 0; i < stack.length; i++) {
    x += parseInt(stack[i]) / 2;
  }
  console.log("DELETE....." + x);
  tasks();
```

Рисунок 2 - Добавление, вывод и удаления узла из стека

```
1-ADD NODE
2-DISPLAY ALL NODES
3-CALCULATE SUM
CHOOSE TASK : 1

ADD ELEMENT : 425

1-ADD NODE
2-DISPLAY ALL NODES
3-CALCULATE SUM
CHOOSE TASK : 2

DISPLAY :

[ '425' ]

1-ADD NODE
2-DISPLAY ALL NODES
3-CALCULATE SUM
CHOOSE TASK : 1

ADD ELEMENT : 63

1-ADD NODE
2-DISPLAY ALL NODES
3-CALCULATE SUM
CHOOSE TASK : 3
DELETE......244

1-ADD NODE
2-DISPLAY ALL NODES
3-CALCULATE SUM
CHOOSE TASK :
```

Рисунок 3 - Результат работы программы