

Лабораторная работа № 1

Стеки

Цель работы: изучить и исследовать структуру данных «стек».

Общие сведения

Понятие очереди всем хорошо известно из повседневной жизни. Элементами очереди в общем случае являются заказы на то или иное обслуживание: выбить чек на нужную сумму в кассе магазина, получить нужную информацию в справочном бюро, выполнить очередную операцию по обработке детали на данном станке в автоматической линии и т.д.

В программировании имеется структура данных, которая называется очередь. Эта структура данных используется, например, для моделирования реальных очередей с целью определения их характеристик (средняя длина очереди, время пребывания заказа в очереди и т.п.) при данном законе поступления заказов и дисциплине их обслуживания.

По своему существу очередь является полустатической структурой – с течением времени и длина очереди, и набор образующих ее элементов могут изменяться.

Различают два основных вида очередей, отличающихся по дисциплине обслуживания находящихся в них элементов :

1. При первой из дисциплин заказ, поступивший в очередь первым, выбирается первым для обслуживания (и удаляется из очереди). Эту дисциплину обслуживания принято называть FIFO (First input-First output, т.е. первый пришел – первый ушел). Очередь открыта с обеих сторон.

2. Вторую дисциплину принято называть LIFO (Last input – First output, т.е. последний пришел – первый ушел), при которой на обслуживание первым выбирается тот элемент очереди, который поступил в нее последним. Очередь такого вида в программировании принято называть СТЕКОМ (магазином) – это одна из наиболее употребительных структур данных, которая оказывается весьма удобной при решении различных задач.

В силу указанной дисциплины обслуживания, в стеке доступна единственная его позиция, которая называется ВЕРШИНОЙ стека – эта позиция, в которой находится последний по времени поступления в стек элемент. Когда заносят новый элемент в стек, то он помещается поверх вершины и теперь уже сам находится в вершине стека. Выбрать элемент можно только из вершины стека; при этом выбранный

элемент исключается из стека, а в его вершине оказывается элемент, который был занесен в стек перед выбранным из него элементом (структура с ограниченным доступом к данным).

Операции над стеками:

- PUSH (s , i) – занесение элемента в стек, где s – название стека, i – элемент, который заносится в стек;
- POP (s) – выборка элемента из стека. При выборке элемент помещается в рабочую область памяти, где он используется;
- EMPTY (s) – проверка стека на пустоту (true – пуст, false – не пуст);
- STACKTOP (s) – чтение верхнего элемента без его удаления.

Фрагмент программы создания стека (необходимые процедуры):

```
Program STACK;
const max_st=50;
var  st,st2: array[1..max_st] of integer;
     n:integer;

function empty:boolean; {Проверка стека на наличие элементов в нем}
begin
empty:=n=0
end;

procedure push(a:char); {Поместить элемент в стек}
begin
inc(n);
st[n]:=a;
end;

procedure pop(var a:char); {Извлечь элемент из стека}
begin
a:=st[n];
dec(n);
end;

function full:boolean; {Проверка на переполнение}
begin
```

```
Full:=n=max_st
end;
```

```
procedure stacktop(var a:char); {Узнать верхний элемент}
begin
a:=st[n];
end;
begin {Основная программа}
.....
end.
```

Стек, как динамическая структура данных

Описать стек, содержащий, например, целые числа, можно следующим образом:

```
Type
PStack = ^TStack;
TStack = Record
    Data : Integer;
    Next : PStack;
end;
Var
    Stack : PStack;
```

Если стек пуст, то значение переменной Stack равно nil.

Примеры реализации операций со стеком

Занесение в стек.

```
Var
    x : PStack;
.....
    New(x);
    x^.Data := .....;
    x^.Next := Stack;
    Stack := x;
.....
```

Извлечение элемента из стека.

В результате выполнения этой операции некоторой переменной N должно быть присвоено значение первого элемента стека и изменено значение указателя на начало стека:

Var

N : Integer;

x : PStack;

.....

N := Stack^.Data;

x:= Stack;

Stack := Stack^.Next;

Dispose(x);

.....

Порядок работы:

1. Изучить описание лабораторной работы.
2. По заданию (в соответствии с вариантом) разработать алгоритм решения задачи (должно быть два алгоритма:
 - 1) с использованием статических структур данных;
 - 2) с использованием динамических структур данных.
3. Разработать проект в среде Delphi.
4. Отладить программу.
5. Оформить отчет.

Проект должен обеспечивать:

1. Ввод данных и помещение их в стек.
2. Преобразование данных в стеке (в соответствии с вариантом).
3. Вывод полученного стека на экран.

Варианты:

1. Поменять местами первый и последний элементы стека.
2. Развернуть стек, т.е. «дно» стека сделать вершиной, а вершину – «дном».
3. Удалить каждый второй элемент стека.
4. Вставить символ '*' в середину стека, если четное число элементов, а если нечетное, то после среднего элемента.
5. Найти максимальный элемент и вставить после него 0.
6. Удалить минимальный элемент.
7. Удалить все элементы, равные первому.
8. Удалить все элементы, равные последнему.
9. Удалить максимальный элемент.
10. Найти минимальный элемент и вставить на его место 0.

Содержание отчета

- а) Титульный лист;
- б) Задание;
- в) Теоретические положения (описание структуры данных «стек»);
- г) Текст проекта (программы);
- д) Результаты работы программы;
- е) Список использованных источников.

Отчет оформляется в электронном виде с обязательным приложением проекта.

Контрольные вопросы

1. Что такое стек?
2. Как можно хранить стек?
3. В чем состоит удобство использования стека при решении задач?
4. Что такое вершина стека?
5. Назовите операции над стеками.
6. Опишите операцию занесения элемента в стек.
7. Опишите операцию выборки элемента из стека.
8. Сравните статические и динамические варианты реализации стека.

Контрольные задания

1. Разработайте программу для удаления минимального элемента из стека.
2. Разработайте программу для удаления из стека всех элементов, равных первому.
3. Разработайте программу для удаления из стека всех элементов, равных последнему.