

Семинар 1 (8.02.17)

1. Преподаватель: Шмелёва Дарья Викторовна (dshmeljowa@yandex.ru)
2. Куртки == плохо

Лекция 1 (9.02.17)

1. Преподаватель: Лобанов Александр Анатольевич
2. Список литературы:
 - (a) Ключаров А.А. «Структуры и алгоритмы обработки данных» СПб-2003г.
 - (b) Пышкин Е.В. «Структуры данных и алгоритмы; реализация на C / C++» СПб-2009г.
3. Алгоритм — точное предписание, определяющее вычислительный процесс от начальных данных к искомому результату.
4. Структура данных — множество элементов данных и связей между ними
 - (a) Внутренние
 - i. Элементарные: числовые, символьные, булевы, указательные
 - ii. Составные
 - A. Линейные: массивы, записи, множества, стеки (stack), очереди, деки
 - B. Нелинейные: деревья, графы, мульти списки
 - (b) Внешние: файлы, базы данных
5. Изменичивость структуры данных — изменение числа элементов и/или связей между составными частями структуры.
 - (a) Статические: векторы, массивы
 - (b) Динамические: линейные связные списки, разветвлённые связные списки, графы, деревья
 - (c) Полустатические: стеки (stack), очереди, деки, строки

6. Массив — структура данных, характеризующаяся фиксированным числом элементов одного типа, каждый из которых имеет уникальный набор значения и индекса.
7. Запись — конечное, упорядоченное множество полей, характеризующихся различным типом данных.
8. Стек — последовательность элементов с переменной длиной. Включение и исключение элементов в нём происходит только с одной стороны, которая называется **вершиной**. (LIFO — Last-In, First-Out)

```
struct stack {  
    char *data;  
    struct stack *next;  
}
```

(a) Операции:

- i. Добавление элемента (push)
- ii. Удаление (исключение) элемента (pop)
- iii. Чтение головного элемента (pick)

(b) Область применения:

- i. обход структур данных
- ii. отслеживание точек возврата из подпрограмм
- iii. сохранение незаконченной работы при переключении на другую задачу

9. Очередь — структура данных FIFO (First-In, First-Out).

```
typedef struct {  
    int qh, qt;  
    int val[MAXN];  
} queue;  
queue a,b;
```

- (a) Физически очередь представляется в памяти аналогично стеку, т.е. в виде вектора, размер которого определяет длину очереди.
- (b) Дескриптор очереди дополнительно содержит два указателя: на начало очереди и на конец этой очереди.

10. Двусвязная очередь (Дек, double-ended queue) — структура данных, в которой элементы можно добавлять и удалять как в начале, так и в конце.

(a) Операции:

- i. Push Back
- ii. Push Front
- iii. Pop Front
- iv. Pop Back

Лекция 2 (16.02.17) «Динамические структуры»

— структуры данных, память под которые выделяется и освобождается по мере необходимости.

1. Если при создании структуры число элементов и характер связи между ними неизвестны, используется метод динамического распределения памяти. Выделяется фиксированный объём памяти для хранения адреса динамического элемента (а не самого элемента).
2. Каждой динамической структуре сопоставляется указатель (адрес), посредством которого осуществляется доступ к динамической структуре.
3. Особенности динамических структур данных:
 - (a) не имеет имени
 - (b) память выделяется в процессе выполнения программы, а не компиляции
 - (c) количество элементов может меняться
 - (d) размерность структуры может меняться
 - (e) характер взаимосвязи между элементами может меняться в процесс выполнения программы
4. Необходимость в динамических структурах возникает в следующих случаях:
 - (a) используются переменные, имеющие довольно большой размер

- (b) в процессе работы программы нужен массив, список и т.д., размер которого изменяется в широких пределах и трудно предсказуем
- (c) размер данных превышает объём сегмента данных

5. Достоинства связного представления данных

- (a) размер структуры ограничивается только объёмом памяти
- (b) при изменении логической структуры не требуется перемещение данных в памяти
- (c) большая гибкость структуры

6. Недостатки:

- (a) на поля, содержащие указатели, расходуется дополнительная память
- (b) доступ к элементам может быть менее эффективным по времени

7. Элемент:
- | |
|---------------|
| P — указатель |
| D — данные |