

Динамические структуры данных

Алексей Владыкин

СПбГУ ИТМО

23 ноября 2009

Абстрактные типы данных

- Вектор (список)
- Стек (магазин, LIFO)
- Очередь (FIFO)
- Дек
- Очередь с приоритетом

Вектор (или список)

```
public interface Vector<T> {  
    T get(int index);  
    void set(int index, T element);  
    void remove(int index);  
    void insert(int index, T element);  
    int size();  
}
```

Реализации:

- массив;
- односвязный или двусвязный список.

Стек (магазин, LIFO)

```
public interface Stack<T> {  
    void push(T element);  
    T pop();  
    int size();  
}
```

Реализации:

- массив;
- СВЯЗНЫЙ СПИСОК.

Применения стека

- В архитектуре современных процессоров: стек вызовов, стек регистров сопроцессора
- В алгоритмах: например, обход графа в глубину
- В языках программирования: например, PostScript

Очередь (FIFO)

```
public interface Queue<T> {  
    void push(T element);  
    T pop();  
    int size();  
}
```

Реализации:

- кольцевой буфер;
- связный список;
- два стека.

Применения очереди

- В «железе»: клавиатурный буфер BIOS
- В программных системах: очередь сообщений (Windows, AWT)
- В алгоритмах: например, обход графа в ширину

```
public interface Deque<T> {  
    void push_front(T element);  
    void push_back(T element);  
    T pop_front();  
    T pop_back();  
    int size();  
}
```

Реализации:

- кольцевой буфер;
- связный список;
- два стека.

Очередь с приоритетом

```
public interface PriorityQueue<T> {  
    void push(T element, int priority);  
    T pop();  
    int size();  
}
```

Реализации:

- связный список;
- куча.