

РЕАЛИЗАЦИЯ АЛГОРИТМОВ

1.1 Эффективные алгоритмы реализации АТД сисок, стек, очередь

Абстрактный тип данных (АТД) — это математическая модель для типов данных, где тип данных определяется поведением (семантикой) с точки зрения пользователя данных, а именно в терминах возможных значений, возможных операций над данными этого типа и поведения этих операций.

В программировании абстрактные типы данных обычно представляются в виде интерфейсов, которые скрывают соответствующие реализации типов. Программисты работают с абстрактными типами данных исключительно через их интерфейсы, поскольку реализация может в будущем измениться. Такой подход соответствует принципу инкапсуляции в объектно-ориентированном программировании. Сильной стороной этой методики является именно сокрытие реализации.

Абстрактные типы данных позволяют достичь модульности программных продуктов и иметь несколько альтернативных взаимозаменяемых реализаций отдельного модуля.

Список — это абстрактный тип данных, представляющий собой упорядоченный набор значений, в котором некоторое значение может встречаться более одного раза. Экземпляр списка является компьютерной реализацией математического понятия конечной последовательности. Экземпляры значений, находящихся в списке, называются элементами списка; если значение встречается несколько раз, каждое вхождение считается отдельным элементом. В языке Lisp список — основная структура данных, как и во многих других функциональных языках.

Стек — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»).

Очередь — абстрактный тип данных с дисциплиной доступа к элементам «первый пришёл — первый вышел» (FIFO, First In — First Out). Добавление элемента (принято обозначать словом enqueue — поставить в очередь) возможно лишь в конец очереди, выборка — только из начала очереди (что принято называть словом dequeue — убрать из очереди), при этом выбранный элемент из очереди удаляется.

1.2 Операции над АТД

Со *стеком* возможны три операции: добавление элемента (иначе проталкивание, push), удаление элемента (pop) чтение головного элемента (peek). В *очереди* можно добавить элемент (только в конец) и извлечь элемент (только с начала). В зависимости от вида и способа реализации *списка* может быть или отсутствовать возможность иметь доступ к произвольному элементу и вставлять/удалять элемент в произвольную позицию списка.