

Абстрактные типы данных

Концепция типа данных

- Любой тип данных определяет множество значений, к которым может относиться некоторая константа, которое может принимать переменная или выражение и которое может формироваться операцией или функцией
- Тип любой величины может быть выведен по ее виду или по ее описанию
- Каждая операция или функция требует аргументов определенного вида и дает результат фиксированного типа

Типы данных

- Простейшие типы данных
- задаются простым перечислением значений
- Простейшие стандартные типы данных
- типы имеющиеся на большинстве вычислительных машин
- Ограниченные типы данных
(диапазоны) – значения некоторого типа, лежащие только внутри определенного интервала

Структуры данных

- Структура данных – это конструкция, определенная в языке программирования для хранения набора данных
- Наиболее распространенные структуры данных:
 - массивы
 - структуры (записи)

Составные типы данных

- Массивы – упорядоченный набор данных одного и того же типа, называемого базовым
- Структуры (записи) – объединение элементов произвольного типа

Общее свойство: массивы и записи обладают *случайным доступом к компонентам*

Абстракция данных

Абстракция данных

- - описывает «Что» можно делать с набором данных, игнорируя вопрос «как это делается?»
- - позволяет разрабатывать отдельные структуры данных независимо от остальной части программы

Абстрактные типы данных

Абстрактный тип данных – это совокупность данных и операций над ними

Структура данных является частью реализации АД

Перед реализацией АД необходимо тщательно описать все операции, которые необходимо выполнять

Абстрактные типы данных

- Списки
- Очереди
- Деревья

Операции над абстрактным Списком

- ***CreateList(List)*** - создает пустой список *List*
- ***DeleteList(List)*** – уничтожает список *List*
- ***IsEmpty(List)*** – определяет пуст ли список *List*
- ***Insert(index, NewElement, List)*** - вставляет новый элемент *NewElement* в список *List* на позицию *index*
- ***Remove(index, List)*** – удаляет элемент списка, находящийся в позиции *index*
- ***Retrive(index, List)*** – возвращает элемент, находящийся в позиции *index*
- ***Getlength(List)*** – возвращает количество элементов в списке *List*

Пример:

```
CreateList(S);  
Insert(1, ПР, S);  
Insert(2, ГТ, S);  
Insert(3, МЛ, S);  
Insert(4, ДМ, S);  
N=GetLength(S);  
For (I=1; I<=n; I++)  
  { Retrive(I, data, s);  
    ВЫВОД на печать элемента data  
  }
```