

Материалы презентации предназначены для размещения только для использования студентами кафедры «Компьютерные системы и технологии» НИЯУ МИФИ дневного и вечернего отделений, изучающими курс «Программирование (Алгоритмы и структуры данных)».

Публикация (размещение) данных материалов полностью или частично в электронном или печатном виде в любых других открытых или закрытых изданиях (ресурсах), а также использование их для целей, не связанных с учебным процессом в рамках курса «Программирование (Алгоритмы и структуры данных)» кафедры «КСиТ» НИЯУ МИФИ, без письменного разрешения автора запрещена.

Введение

Методологии программирования

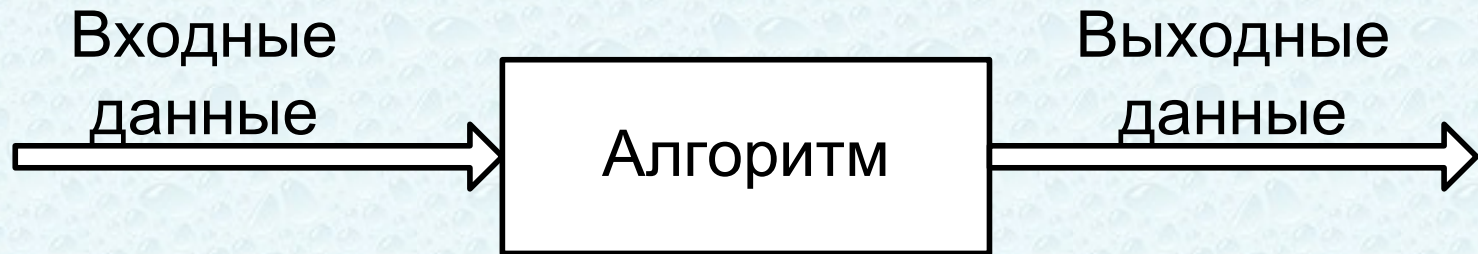
- Хаотическое программирование
- Структурное программирование
- Объектно-ориентированное программирование

Алгоритм

- Программа = алгоритм + данные (Вирт)
- Алгоритм
 - любая корректно определенная вычислительная процедура, на вход которой подается некоторая величина или набор величин, и результатом выполнения которой является выходная величина или набор значений

Алгоритм

- Программа = алгоритм + данные (Вирт)
- Алгоритм



Пример: задача сортировки

Вход: $\{ a_1, a_2, \dots, a_n \}$

Выход: $\{ a_1', a_2', \dots, a_n' \}, a_1' \leq a_2' \leq \dots \leq a_n'$

Алгоритм

- Эффективность алгоритма $t = k / v$

1. $v_1 = 10^9$ оп/сек;

сортировка вставками; $k_1 = c_1 n^2$; $c_1 = 2$

2. $v_1 = 10^7$ оп/сек;

сортировка слиянием; $k_1 = c_2 n \log_2(n)$; $c_2 = 50$

n	t₁	t₂
10^6	2000 сек	≈ 100 сек
10^7	$\approx 2,3$ дня	< 20 мин

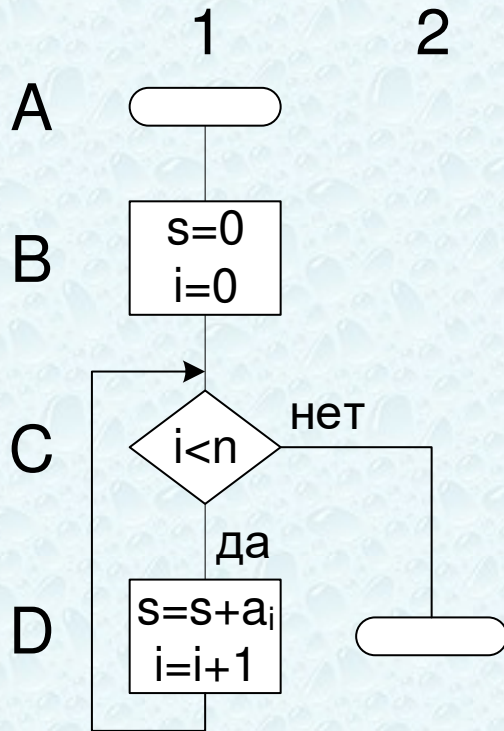
Алгоритм

Представление алгоритма

- словесное описание
- схема алгоритма
- псевдокод
- программа

Алгоритм

Схема алгоритма



Псевдокод

$s = 0, i = 0$

```
while не все элементы {  
    добавить к s очередной  
    элемент:  $s = s + a_i$   
     $i = i + 1$   
}
```

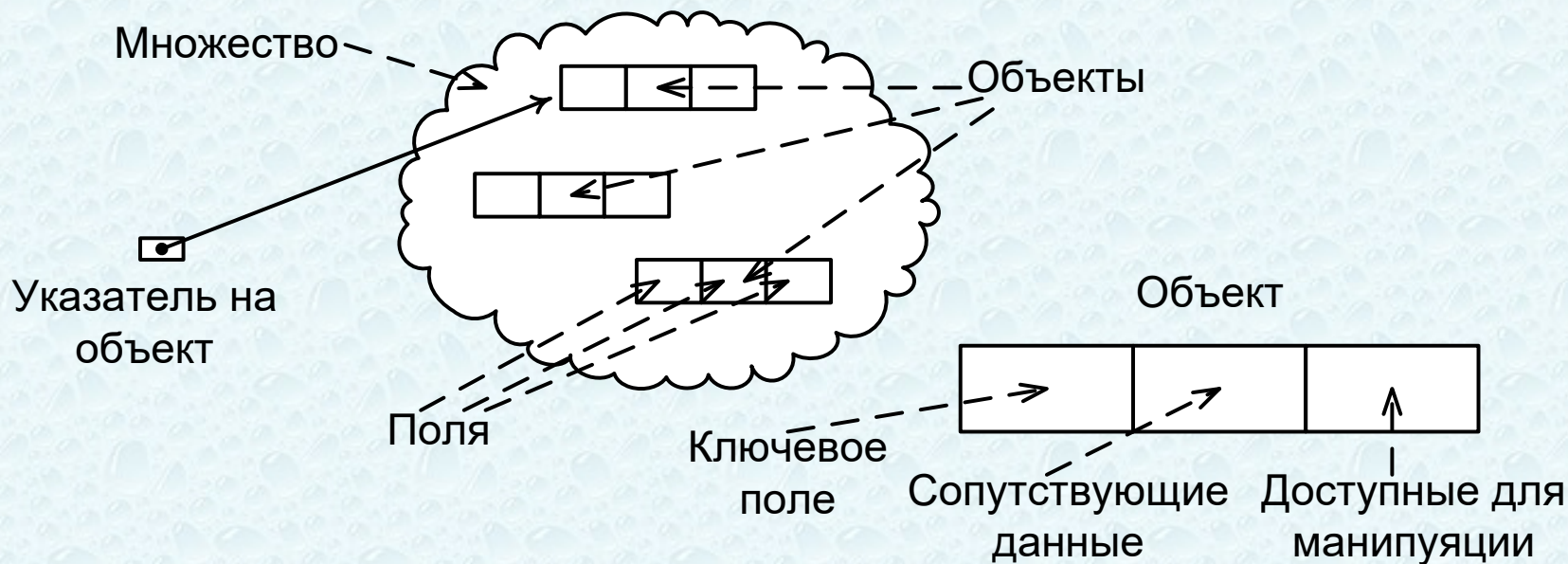

1. Общая характеристика структур данных

Уровни представления информации

- Интуитивные структуры
- Абстрактные (логические) структуры
- Конкретные структуры

Структура данных

- Логическая структура данных = множество элементов + связи между элементами
- Динамическое множество



Логические структуры

- Массивы
- Строки
- Стеки, очереди, деки
- Таблицы
- Деревья
- Графы

Операции

Запросы	Модифицирующие операции
Search (S, k)	Insert (S, x)
Minimum (S)	Delete (S, x)
Maximum (S)	
Successor (S, x)	
Predecessor (S, x)	

S – динамическое множество

x – указатель на элемент множества

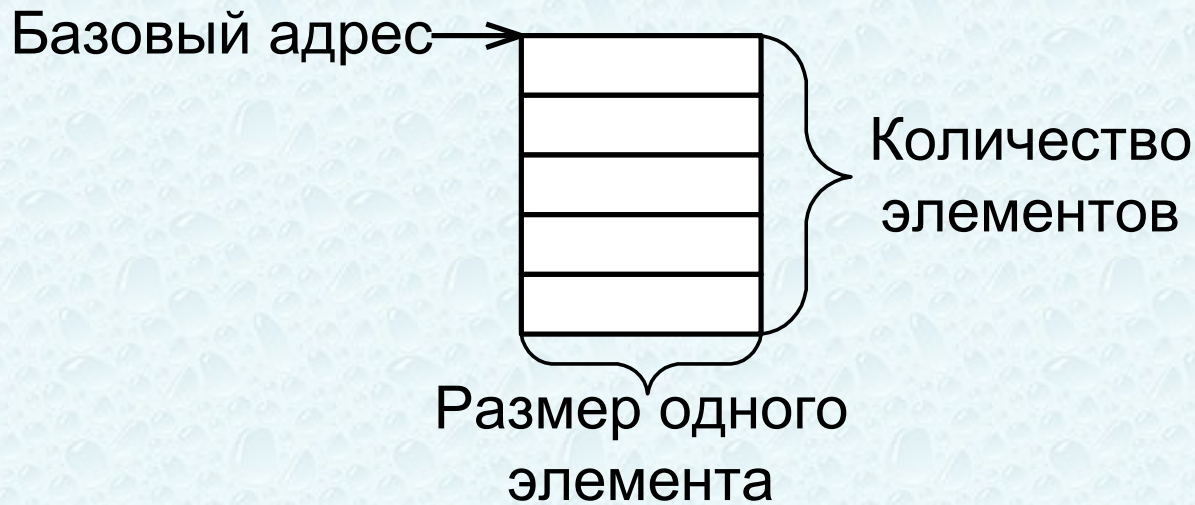
k – значение ключа

Конкретные структуры

Отображение в памяти ЭВМ

- Вектор
- Список
- Сеть

Вектор



Доступ к элементу – по индексу

$$\text{Адрес}_i = \text{базовый_адрес} + i * \text{размер_элемента}$$

Список

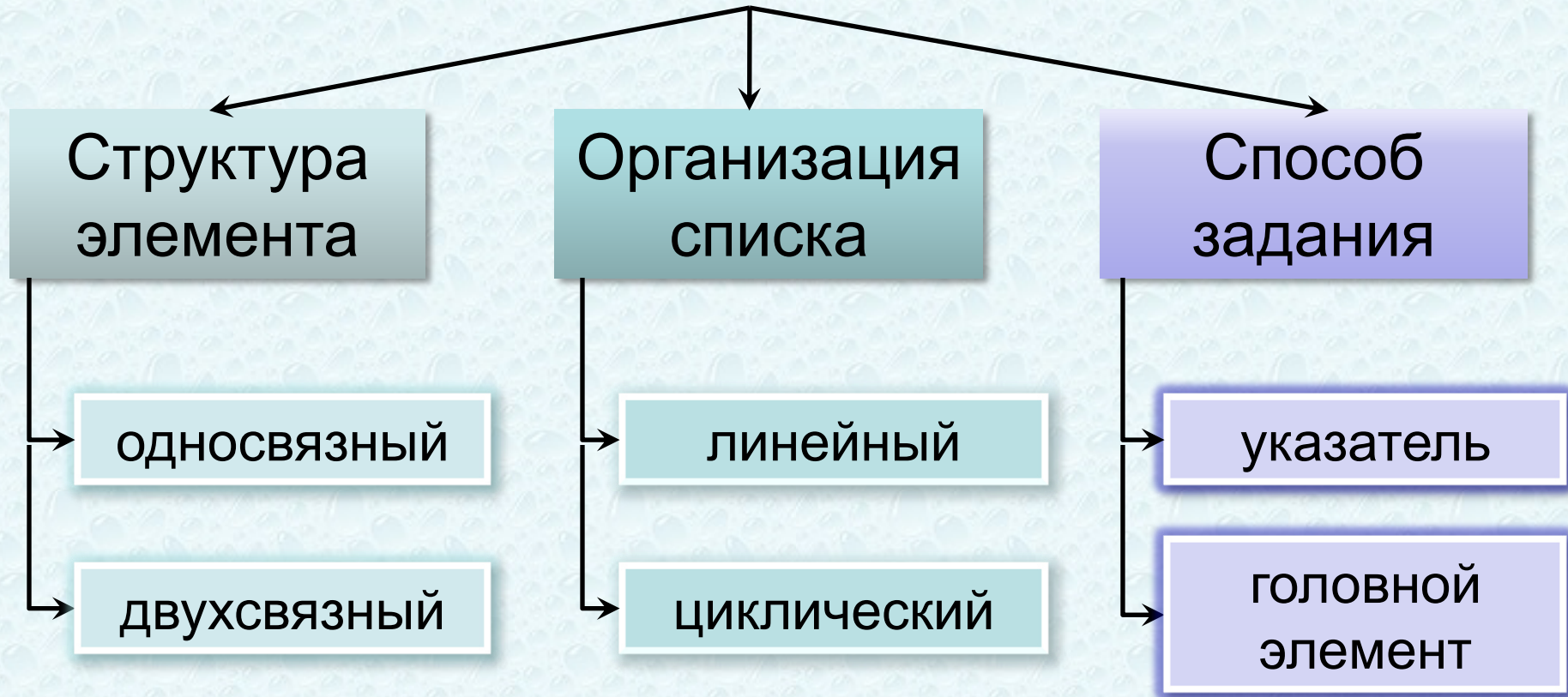
Элемент списка

Информация	Поле связи
------------	------------

Доступ к элементу – по указателю

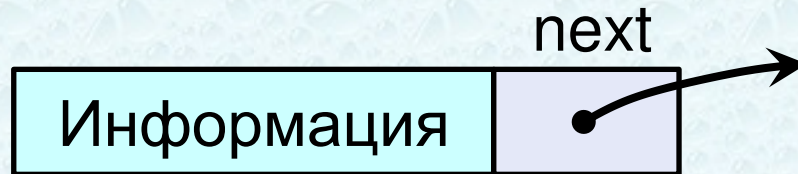
Список

Типы списков

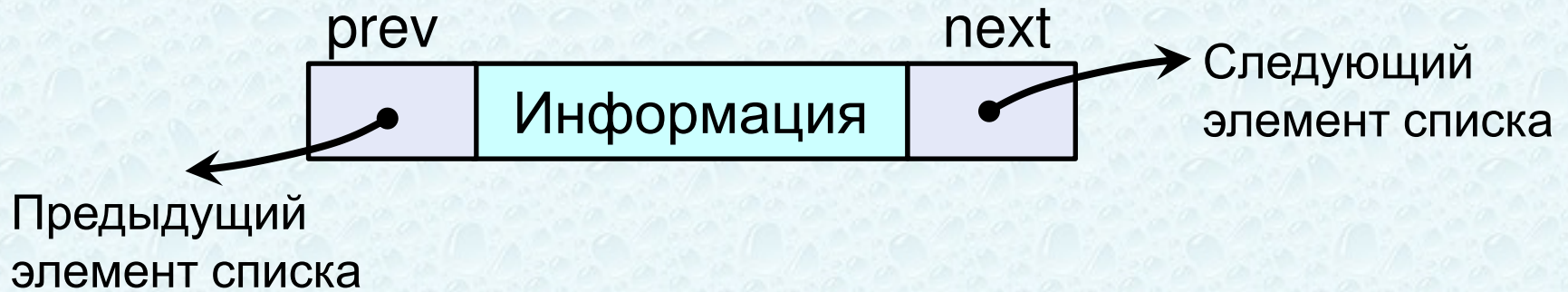


Элемент списка

- Односвязный список



- Двухсвязный список

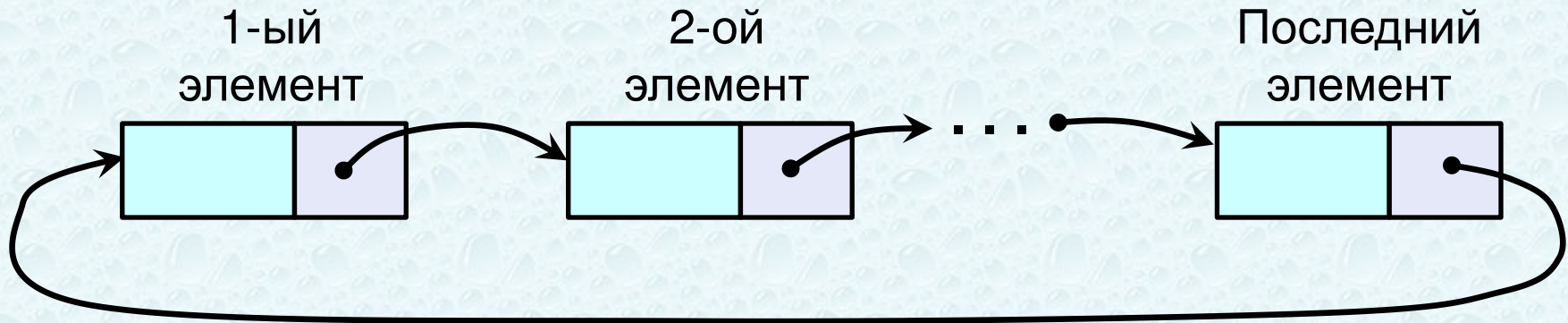


Организация списка

- Лине́йный список

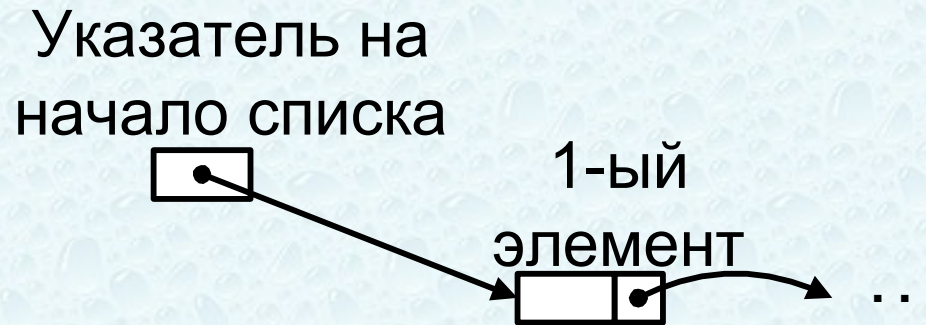


- Циклический список



Задание списка

- Указатель на начало списка



- Головной элемент

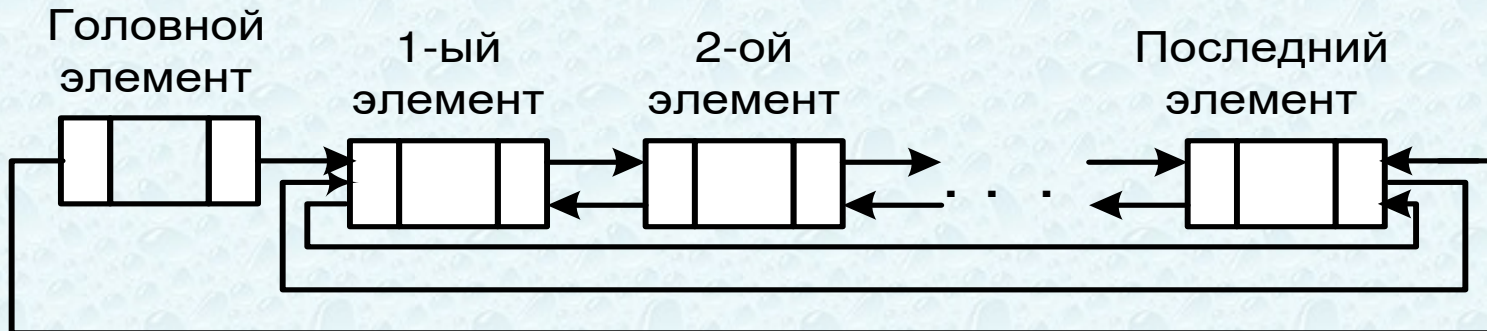


Примеры

Линейный односвязный список с головным элементом



Циклический двухсвязный список с головным элементом



Сеть

