

Структуры и алгоритмы обработки данных

Литература:

- **Кнут Д. Искусство программирования. т.1,2,3,4. Третье издание. - Вильямс, 2010. (1968, 1969, 1973, 2005)**
- **Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы. - ИД Вильямс, 2003.**
- **Вирт Н. Алгоритмы + структуры данных = программы. – Невский диалект, 2008. (1985)**
- **Вирт Н. Алгоритмы и структуры данных. – Невский диалект, 2008. (1989)**
- **Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. - Вильямс, 2010.**
- **Каррано Ф., Причард Дж. Абстракция данных и решение задач на С++. Стены и зеркала, 3-е издание. – Вильямс, 2003. (2002)**
- **Кубенский А.А. Структуры и алгоритмы обработки данных. Объектно-ориентированный подход и реализация на С++. – БХВ-Петербург, 2004.**
- **Гудрич М., Томасия Р. Структуры данных и алгоритмы в Java. – Минск: Новое знание, 2003. (2001)**

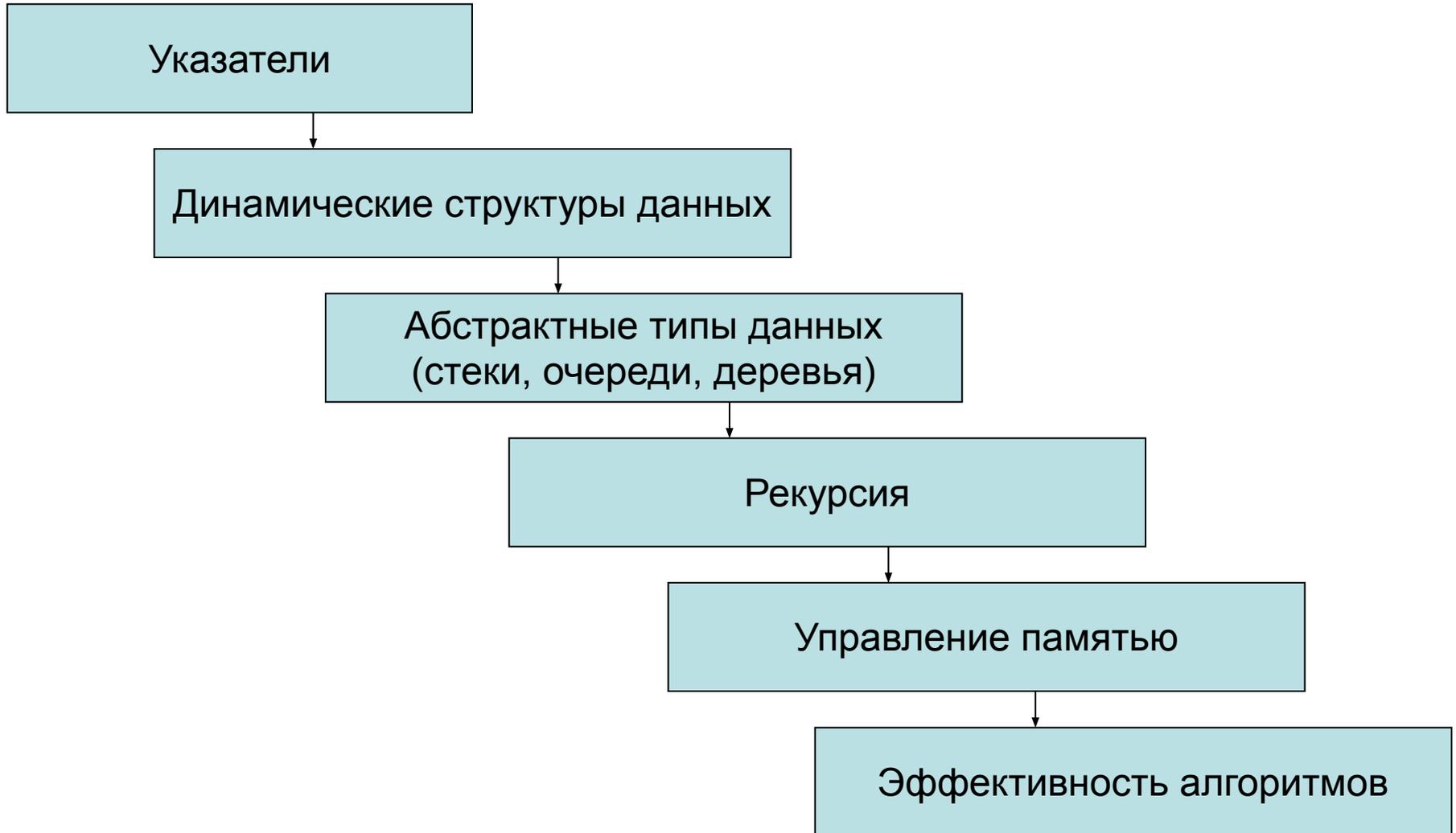
Цели и задачи курса

Цель – научиться технике конструирования и формулирования алгоритмов, описывающих некоторые типовые процессы обработки данных.

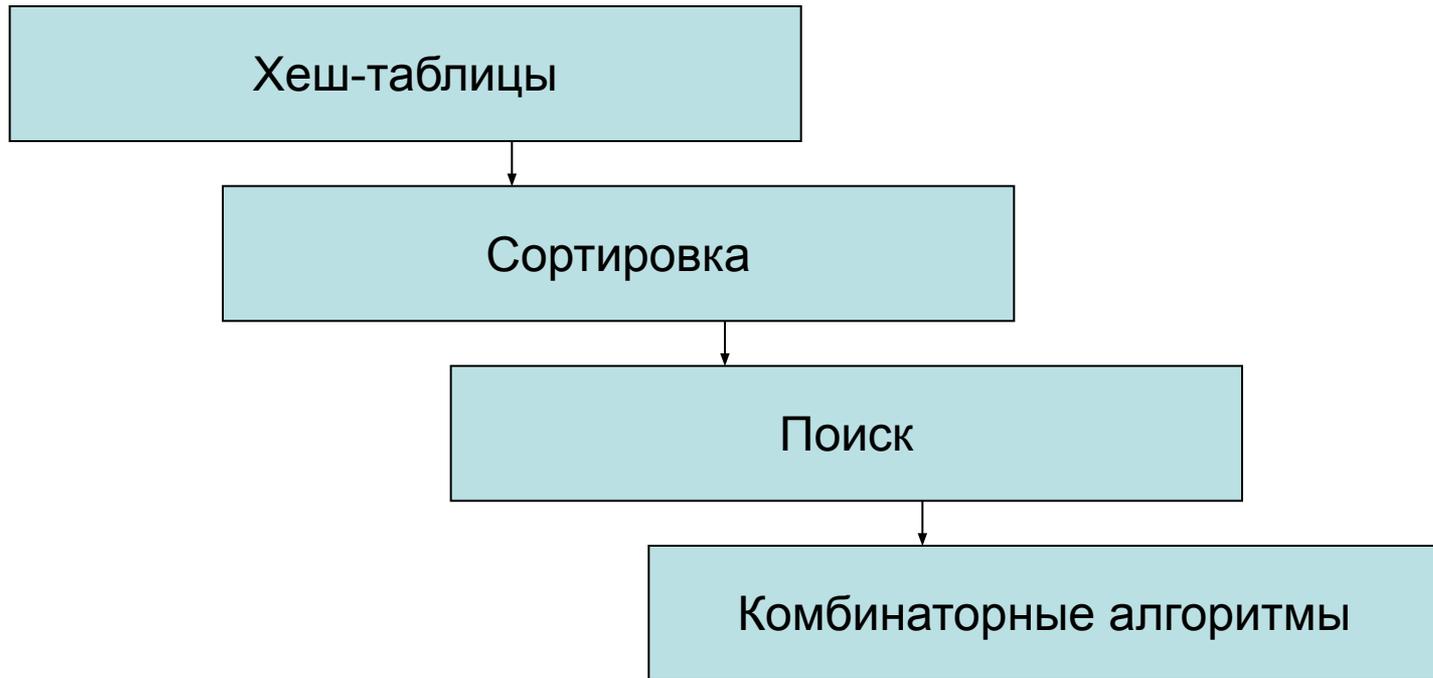
Задачи:

1. Алгоритмы и структуры данных взаимосвязаны.
2. Использовать приемы, не зависящие от конкретных приложений.
3. Язык инструмент, а не самоцель.
4. Абстрактные типы данных и структуры данных — не одно и то же.
5. От сложного к простому через рекурсию.
6. Оценка эффективности алгоритмов – составная часть компьютерного решения задач.

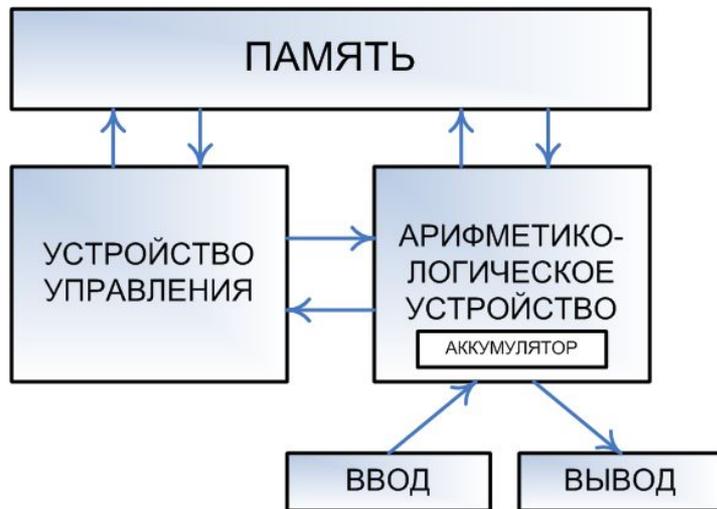
Структура курса (1 часть)



Структура курса (2 часть)



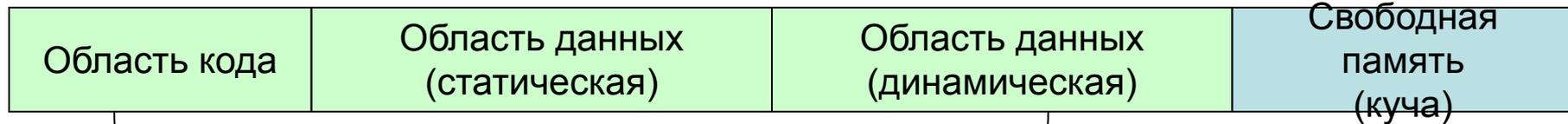
Основные понятия - память



Фон-Неймановская архитектура

Однородность памяти

Куча



Память программы

Основные понятия - данные

тип данных - множество значений, которые может принимать переменная

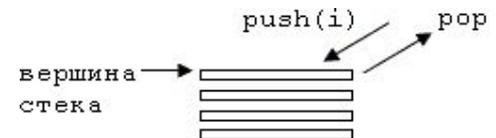
```
int a;           // объявление переменной a целого типа
float b;         // объявление переменной с плавающей запятой
char d = 's';    // инициализация переменной типа char
```

структура данных - набор переменных, возможно, различных типов данных, объединенных определенным образом (способ организации данных)

```
struct str_name
{
    int member_1;
    float member_2;
};
```

Абстрактный тип данных - это тип данных, который предоставляет для работы с его элементами определённый набор функций.

```
class Stack
{
    .....
    void Push(int i);
    int Pop();
    bool isEmpty();
    .....
};
```



Основные понятия - алгоритм

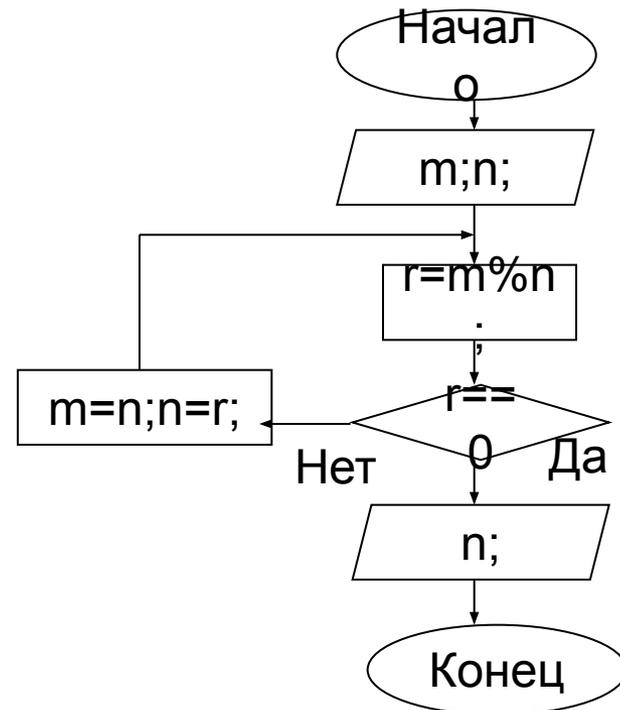
Алгоритм - точная конечная последовательность действий, обеспечивающая получение требуемого результата из заданных исходных данных.

Основные свойства алгоритмов:

1. Дискретность
2. Определенность (детерминированность)
3. Результативность
4. Массовость
5. Эффективность

Алгоритм Евклида.

Найти наибольший общий делитель для целых $m > n$.



**ГОСТ 19.701—90
(ИСО 5807—85)**

М Е Ж Г О С У Д А Р С Т В Е Н Н Ы Й С Т А Н Д А Р Т

**ЕДИНАЯ СИСТЕМА
ПРОГРАММНОЙ ДОКУМЕНТАЦИИ**

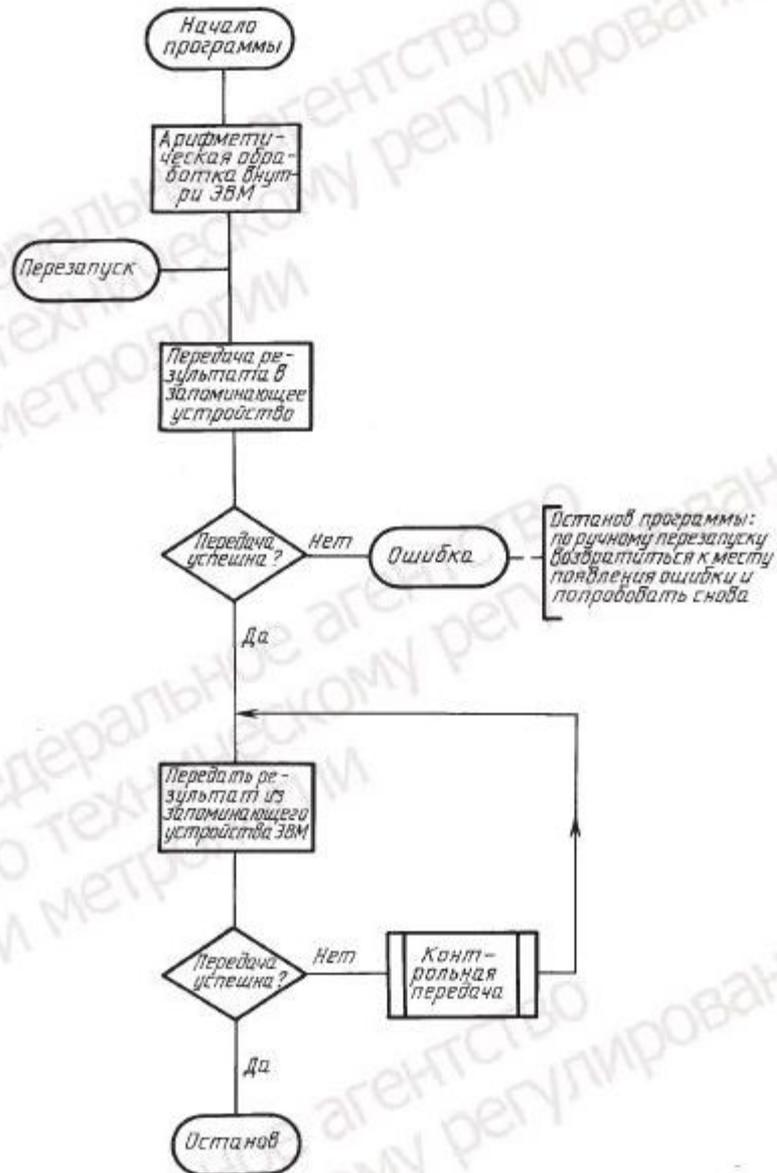
**СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ,
ДАНЫХ И СИСТЕМ**

**ОБОЗНАЧЕНИЯ УСЛОВНЫЕ
И ПРАВИЛА ВЫПОЛНЕНИЯ**

Издание официальное

2. Схемы программы

Пример 1.



3.2.2.2. Ручная операция

Символ отображает любой процесс, выполняемый человеком.



3.2.2.3. Подготовка

Символ отображает модификацию команды или группы команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация индексного регистра или инициализация программы).



3.2.2.4. Решение

Символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа. Соответствующие результаты вычисления могут быть записаны по соседству с линиями, отображающими эти пути.

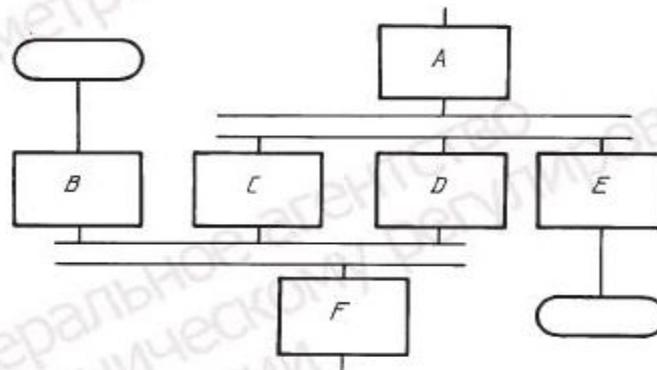


3.2.2.5. Параллельные действия

Символ отображает синхронизацию двух или более параллельных операций.



Пример.



Примечание. Процессы C, D и E не могут начаться до тех пор, пока не завершится процесс A; аналогично процесс F должен ожидать завершения процессов B, C и D, однако процесс C может начаться и (или) завершиться прежде, чем соответственно начнется и (или) завершится процесс D.

3. ОПИСАНИЕ СИМВОЛОВ

3.1. Символы данных

3.1.1. Основные символы данных

3.1.1.1. Данные

Символ отображает данные, носитель данных не определен.



3.1.1.2. Запоминаемые данные

Символ отображает хранимые данные в виде, пригодном для обработки, носитель данных не определен.



3.1.2. Специфические символы данных

3.1.2.1. Оперативное запоминающее устройство

Символ отображает данные, хранящиеся в оперативном запоминающем устройстве.



3.1.2.2. Запоминающее устройство с последовательным доступом

Символ отображает данные, хранящиеся в запоминающем устройстве с последовательным доступом (магнитная лента, кассета с магнитной лентой, магнитофонная кассета).



3.1.2.3. Запоминающее устройство с прямым доступом

Символ отображает данные, хранящиеся в запоминающем устройстве с прямым доступом (магнитный диск, магнитный барабан, гибкий магнитный диск).



3.1.2.4. Документ

Символ отображает данные, представленные на носителе в удобочитаемой форме (машинограмма, документ для оптического или магнитного считывания, микрофильм, рулон ленты с итоговыми данными, бланки ввода данных).

3.4. Специальные символы

3.4.1. Соединитель

Символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте. Соответствующие символы-соединители должны содержать одно и то же уникальное обозначение.



3.4.2. Терминатор

Символ отображает выход во внешнюю среду и вход из внешней среды (начало или конец схемы программы, внешнее использование и источник или пункт назначения данных).

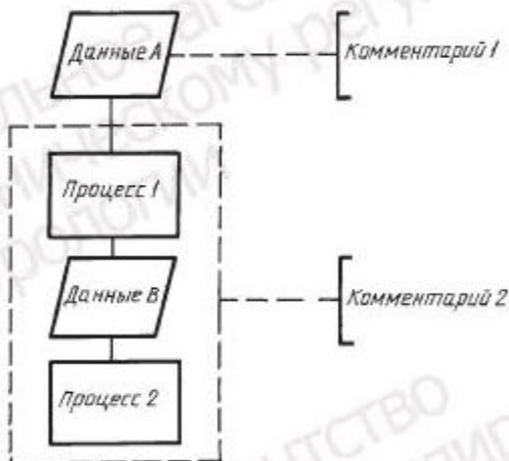


3.4.3. Комментарий

Символ используют для добавления описательных комментариев или пояснительных записей в целях объяснения или примечаний. Пунктирные линии в символе комментария связаны с соответствующим символом или могут обходить группу символов. Текст комментариев или примечаний должен быть помещен около ограничивающей фигуры.



Пример.



3.4.4. Пропуск

Символ (три точки) используют в схемах для отображения пропуска символа или группы символов, в которых не определены ни тип, ни число символов. Символ используют только в символах линии или между ними. Он применяется главным образом в схемах, изображающих общие решения с неизвестным числом повторений.

Примеры.

