

Введение в КОМПЬЮТЕРНЫЕ НАУКИ

ЛЕКТОР К.Т.Н. МОХОВ В.А.

ГЛАВА 5. АЛГОРИТМЫ

Раздел 5: Алгоритмы

- ▶ 5.1 Понятие алгоритма
- ▶ 5.2 Представление алгоритма
- ▶ 5.3 Создание алгоритма
- ▶ 5.4 Итерационные структуры
- ▶ 5.5 Рекурсивные структуры
- ▶ 5.6 Эффективность и правильность

Определение алгоритма

Алгоритм – это **упорядоченный** набор из **недвусмысленных** и **выполнимых** этапов, определяющий некоторый **конечный** процесс.

Представление алгоритма

- ▶ Требует четко определенных примитивов
- ▶ Коллекция примитивов представляет собой язык программирования.

Рисунок 5.2 Как сложить птичку из квадратного листа бумаги

5-5

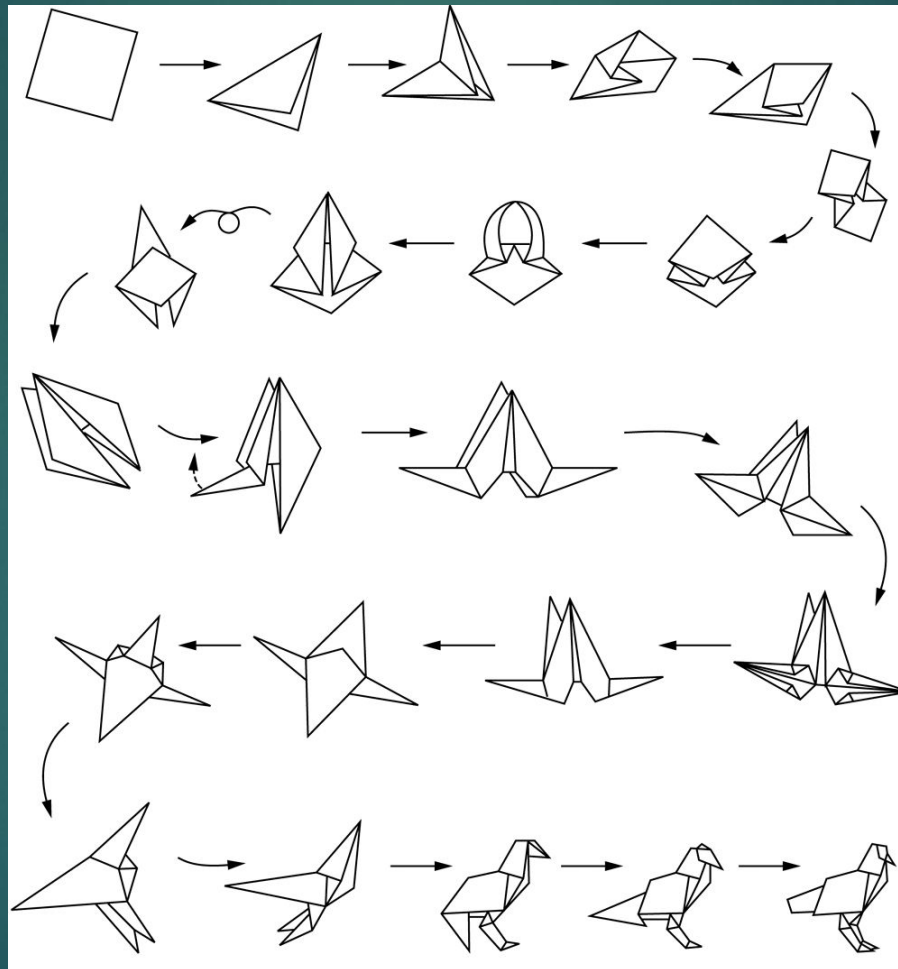


Рисунок 5.3 Примитивы оригами

5-6

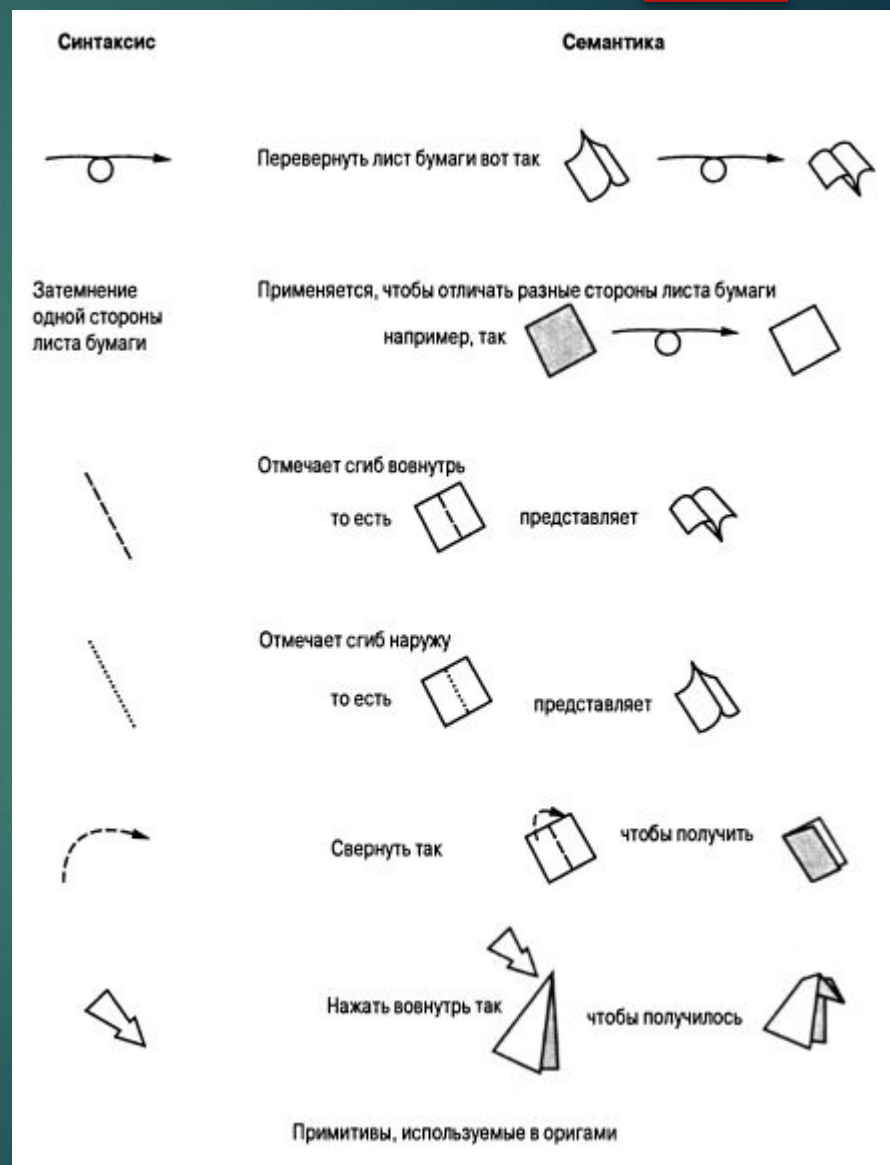
Примитив = Синтаксис + Семантика

Семантика

определяет символическое
представление примитива

Синтаксис

определяет значение примитива



Примитивы псевдокода

5-7

▶ Присваивание

(имя) □ (выражение)

▶ Условный выбор

if *(условие)* **then** *(действие)*

Примитивы псевдокода (продолжение)

- ▶ Цикл с предусловием
while (условие) **do** (действие)

- ▶ Процедура
procedure имя (параметры)

Рисунок 5.4 Процедура приветствия в псевдокоде

```
procedure Greetings  
Count  $\leftarrow$  3;  
while (Count > 0) do  
    (print the message "Hello" and  
     Count  $\leftarrow$  Count -1)
```

Шаги общего плана решения проблемы

- ▶ 1. Понять проблему.
- ▶ 2. Разработать план решения задачи.
- ▶ 3. Осуществить свой замысел.
- ▶ 4. Оценить точность решения и его потенциал, как инструмента для решения других проблем

ПИНОК В ДВЕРЬ

- ▶ Попробуйте решить проблему с конца
- ▶ Облегчите решение связанных задач
 - ▶ Отбросьте некоторые проблемные ограничения
 - ▶ Решите сначала части проблемы (методология «снизу вверх»)
- ▶ Используйте поэтапное уточнение: разделите проблему на более мелкие проблемы (методология «сверху вниз»)

Проблема возраста детей

5-12

- ▶ Некто А хочет определить возраст троих детей некоего В
 - ▶ В сообщает А, что произведение возрастов его детей равно 36.
 - ▶ А сообщает, что нужна ещё подсказка.
 - ▶ В сообщает А сумму возрастов детей.
 - ▶ А повторяет, что ему нужна ещё подсказка.
 - ▶ В говорит А, что старший из детей играет на пианино.
 - ▶ А сообщает В возраст всех трёх детей..
- ▶ Сколько лет детям?

Рисунок 5.5

(1, 1, 36) (1, 6, 6)
(1, 2, 18) (2, 2, 9)
(1, 3, 12) (2, 3, 6)
(1, 4, 9) (3, 3, 4)

$$\begin{aligned}1 + 1 + 36 &= 38 \\1 + 2 + 18 &= 21 \\1 + 3 + 12 &= 16 \\1 + 4 + 9 &= 14\end{aligned}$$

$$\begin{aligned}1 + 6 + 6 &= 13 \\2 + 2 + 9 &= 13 \\2 + 3 + 6 &= 11 \\3 + 3 + 4 &= 10\end{aligned}$$

а) Тройки чисел, произведение которых равно 36

б) Суммы троек чисел, приведенных в а)

Итерационные структуры

- ▶ Цикл с предусловием:

```
while (условие) do  
    (тело цикла)
```

- ▶ Цикл с постусловием:

```
repeat (тело цикла)  
    until (условие)
```

Рисунок 5.6 Алгоритм последовательного поиска, сформулированный с помощью псевдокода

```
procedure Поиск (<список>, <искомое значение>)  
if (список пуст)  
  then (объявить поиск неудачным)  
  else  
    [выбрать как <проверяемое значение> первый элемент списка.  
    while (<искомое значение> > <проверяемое значение>  
      и есть непроверенные элементы)  
      do (выбрать следующий элемент списка  
        как <проверяемое значение>)  
    if (<искомое значение> = <проверяемое значение>)  
      then (объявить поиск успешным)  
      else (объявить поиск неудачным)  
    ]  
  ]
```

Рисунок 5.7 Операции процедуры управления повторяющимися действиями

- Инициализация:** Установить начальное состояние, которое будет модифицироваться до тех пор, пока не будет выполнено условие окончания
- Проверка:** Сравнить текущее состояние с условием завершения и прекратить повторение тела цикла, если состояние соответствует условию окончания
- Модификация:** Изменить состояние таким образом, чтобы приблизиться к ситуации, в которой будет выполнено условие окончания

Рисунок 5.8 Структура цикла типа while-do

5-17



Рисунок 5.9 Структура цикла типа repeat-until

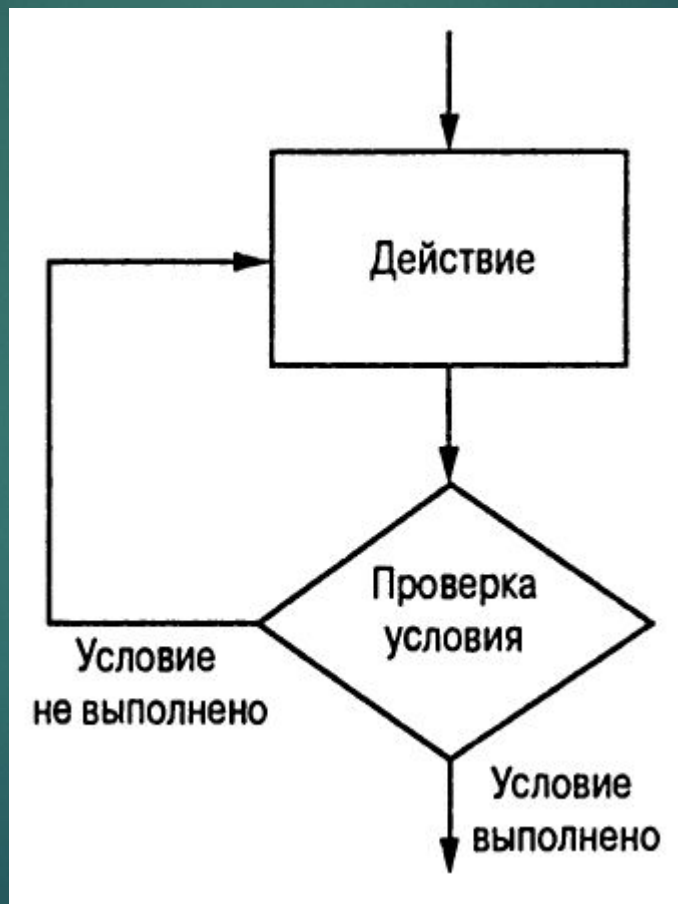


Рисунок 5.10 Сортировка списка имён Fred, Alex, Diana, Byron и Carol в алфавитном порядке

5-19

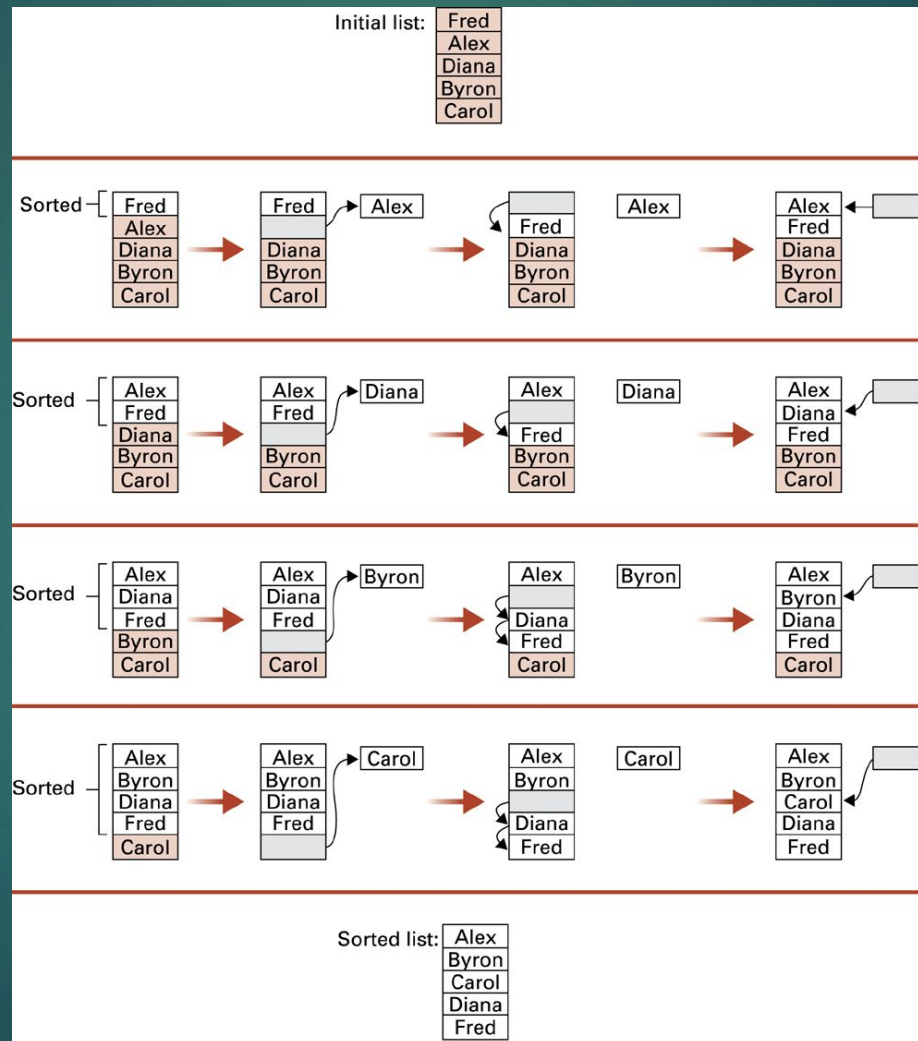


Рисунок 5.11 Алгоритм сортировки методом вставки, написанный в псевдокоде

5-20

```
procedure Сортировка (<список>
assign N the value 2;
while (значение N не превышает <длина списка>) do
  (Выбрать N-й элемент списка в качестве опорного элемента;
  Переместить опорный элемент во временное хранилище,
  оставив в списке пустое место;
  while (над пустым местом есть имя, которое по алфавиту
  размещается ниже, чем опорный элемент) do
    (переместить имя, находящееся над пустым местом вниз,
    оставив в его прежней позиции пустое место)
  Поместить опорный элемент на пустое место в списке
assign N the value N+1
)
```

Algorithm Efficiency

- ▶ Measured as number of instructions executed
- ▶ Big theta notation: Used to represent efficiency classes
 - ▶ Example: Insertion sort is in $\Theta(n^2)$
- ▶ Best, worst, and average case analysis

Классы сложности

5-22



Рисунок 5.18 Работа алгоритма сортировки методом вставки в наихудшем случае



Рисунок 5.19 График продолжительности работы алгоритма сортировки методом вставки для наихудшего случая

5-24



Рисунок 5.20 График продолжительности работы алгоритма двоичного поиска для наихудшего случая

5-25

