

# Алгоритмы и структуры данных

Зариковская Наталья Вячеславовна  
Доцент кафедры ЭМИС

# Введение

- В последние годы программирование для персональных компьютеров сформировалось в некоторую дисциплину, владение которой стало ключевым моментом, определяющим успех многих инженерных проектов, а сама она превратилась в объект научного исследования. Из ремесла программирование перешло в разряд академических наук. Крупные специалисты в области программирования показали, что программы поддаются точному анализу, основанному на строгих математических рассуждениях. При этом можно избежать многих ошибок, традиционных для программистов, если они будут осмысленно пользоваться методами и приемами, которые раньше применялись ими интуитивно. Кроме того программисты уделяли гораздо больше внимания построению и анализу программы, чем выбору представления данных. Современная методология программирования предполагает, что оба аспекта программирования — запись алгоритма на языке программирования и выбор структур представления данных — заслуживают одинакового внимания. Всем, кто занят вопросами программирования для вычислительной техники, известно, что решение о том, как представлять данные, невозможно принимать без понимания того, какие алгоритмы будут к ним применяться, и наоборот, выбор алгоритма часто очень сильно зависит от строения данных, к которым он применяется.

# Введение

- Без знания структур данных и алгоритмов невозможно создать серьезный программный продукт. Ни одна информационная система не обходится без программного обеспечения, она просто не может существовать без этой компоненты. В связи с этим задача данного курса состоит в следующем:
- познакомить со всем разнообразием имеющихся структур данных, показать, как эти структуры реализованы в языках программирования;
- рассмотреть основные операции, которые выполняются над структурами данных;
- показать особенности структурного подхода к разработке алгоритмов, продемонстрировать порядок их разработки.

# Введение

- Без знания структур данных и алгоритмов невозможно создать серьезный программный продукт. Ни одна информационная система не обходится без программного обеспечения, она просто не может существовать без этой компоненты. В связи с этим задача данного курса состоит в следующем:
- познакомить со всем разнообразием имеющихся структур данных, показать, как эти структуры реализованы в языках программирования;
- рассмотреть основные операции, которые выполняются над структурами данных;
- показать особенности структурного подхода к разработке алгоритмов, продемонстрировать порядок их разработки.

# Основные этапы решения задач на ЭВМ

- Решение задачи на ЭВМ сводится к выполнению программы, введённой в память ЭВМ. Решение задачи можно представить в виде следующей последовательности этапов:
- Математическая постановка задачи (формализация).
- Выбор или разработка метода решения (метод решения).
- Разработка алгоритма.
- Написание программы и подготовка её к вводу в ЭВМ (программа).
- Отладка программы и её выполнение (ЭВМ).

# Основные этапы решения задач на ЭВМ

- 1. Математическая постановка задачи и её формализация состоит в определении и уточнении условий и цели задачи. Условие задачи – точное описание исходных данных их характеристик, а также ограничений , записываемых в математической форме.
- Если задача математическая и известны метод её решения и соответствующие этому методу уравнения этап формализации задачи может не потребоваться.

# Основные этапы решения задач на ЭВМ

- 2. Выбор или разработка метода решения тесно связан с этапом формализации, т. к. его цель состоит в сведении задачи к некоторой математической модели, для которой известен метод её решения. Но наиболее интересен случай разработки нового метода.

# Основные этапы решения задач на ЭВМ

- 3. Разработка алгоритма. Алгоритмом называется система правил, чётко описывающая последовательность действий, которые необходимо выполнить для решения задачи.

# Основные этапы решения задач на ЭВМ

- 4. Написание программы и подготовка её к вводу в ЭВМ. Цель этого этапа – запись алгоритма на выбранном языке программирования и переносе этого текста на носитель, с которого она может быть введена в ЭВМ. При выборе языка программирования необходимо пользоваться следующими принципами:
  - а) простота написания программы и сокращения сроков отладки;
  - б) эффективность: затрачиваемое машинное время и требуемый объём памяти.

# Основные этапы решения задач на ЭВМ

- 5. Отладка программы и её выполнение. Цель – выявление и исправление ошибок в программе. Обычно на отладку программист затрачивает от 20 до 40% времени, отводимого на разработку программы.

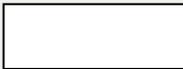
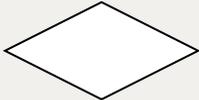
# Основные этапы решения задач на ЭВМ

- Алгоритм – формализованный процесс решения задачи, сведённый к применению конечной последовательности достаточно простых правил.
- Алгоритм обладает следующими основными свойствами: дискретностью, определённостью, результативностью, массовостью.
- **Дискретность** – процесс преобразования исходных данных в результат осуществляется дискретно.
- **Определённость** (детерминированность) – каждое правило алгоритма должно быть чётким и однозначным.
- **Результативность** (конечность) – алгоритм должен приводить к решению задачи за конечное число шагов.
- **Массовость** – алгоритм – решение задачи в общем виде. Задачи могут отличаться значением исходных данных.

# Основные этапы решения задач на ЭВМ

- Запись алгоритма на естественном языке.
- Типичные действия алгоритма записываются следующим образом:
- Этап обработки:  $v = \text{выражение}$ , где  $v$  - переменная.
- Проверка условия: Если Условие идти к  $N$ , где  $N$  метка.
- Переход к этапу с номером  $N$  : Идти к  $N$ .
- Конец вычислений: Остановка

# Изображение алгоритма в виде СХЕМЫ

Процесс		Выполнение операции или группы операций
Решение (ветвление)		Выбор направления выполнения алгоритма или программы в зависимости от выполнения условий
Предопределённый процесс		Использование ранее созданных и отдельно описанных программ (алгоритмов).
Ввод-вывод		Преобразование данных в форму пригодную для обработки.
Линия потока		Линия потока
Соединитель		Указание на связь между прерванными линиями потока.

Программа это последовательность команд, понятных ЭВМ.

# Понятие о структурном подходе к разработке алгоритма.

- Структурный подход к разработке алгоритмов и программ предполагает использование только нескольких основных структур, комбинация которых даёт всё многообразие алгоритмов и программ.
- К основным структурам относятся : 1. Следование; 2. Цикл «До»; 3. Цикл «Пока»; 4. Разветвление; 5. Обход.
- **Следование** – последовательное размещение блоков и групп блоков (последовательное размещение операторов).
- **Цикл «До»** - применяется при необходимости выполнить какие либо вычисления несколько раз до тех пор, пока выполняется некоторое условие. Особенность этого цикла состоит в том, что тело цикла выполняется хотя-бы один раз. Тело цикла – та последовательность операторов в цикле которая выполняется многократно.
- На естественном языке цикл «До» соответствует последовательности операторов.
  - Операторы начальных присваиваний.
  - Операторы тела цикла.

# Понятие о структурном подходе к разработке алгоритма.

**Цикл «Пока».** В отличие от /2/ условие проверяется в начале. На естественном языке цикл «Пока» соответствует последовательности операторов.

1. Операторы начального присваивания
2. Если условие выполняется, то идем к 3, иначе к 5
3. Операторы тела цикла
4. идти к 2
5. продолжить выполнение программы

**4. Разветвление** – применяется, когда в зависимости от условия нужно выполнить либо одно, либо другое действие.

1. Если условие выполняется идти к действию 1
2. Иначе идти к действию 2

# Понятие о структурном подходе к разработке алгоритма.

**Цикл «Пока».** В отличие от /2/ условие проверяется в начале. На естественном языке цикл «Пока» соответствует последовательности операторов.

1. Операторы начального присваивания
2. Если условие выполняется, то идем к 3, иначе к 5
3. Операторы тела цикла
4. идти к 2
5. продолжить выполнение программы

**4. Разветвление** – применяется, когда в зависимости от условия нужно выполнить либо одно, либо другое действие.

1. Если условие выполняется идти к действию 1
2. Иначе идти к действию 2

**5. Обход** – частный случай разветвления, когда одна ветвь не содержит никаких действий.

На естественном языке «Обход»

1. Если условие выполняется, то выполнить действие
2. Иначе просто продолжить программу

# Приёмы разработки алгоритмов.

**1. Метод пошаговой детализации** – при котором первоначально продумывается и фиксируется общая структура алгоритма без детальной проработки отдельных его частей. При этом используются лишь основные структуры алгоритмов. Далее прорабатываются отдельные, не детализированные на предыдущем шаге. Полностью закончив детализацию всех блоков, получаем решение в целом.

**2. Программирование «сверху вниз»** - модуль может быть вызван только теми модулями, которые принадлежат более высокому уровню иерархии. Модульная программа похожа на конструкцию, собранную из блоков, каждый из которых имеет стандартные средства для связи с другими.

**3. Метод разработки программ – «снизу вверх».** В этом методе большое внимание уделяется представлению данных. По этой причине программист должен сначала заняться реализацией средств, а затем перейти к разработке структуры всей программы.

**4. Метод «от центра по краям».** Суть этого метода заключается в том, что сначала выделяется наиболее сложная часть проблемы и ищется её решение, а затем проводятся все оставшиеся работы. В процессе решения проблемы движение происходит как бы от центра по краям.