

---

# Проектирование ПО

---

Лекция 7

Тема 2: Планирование и проектирование ПО

---

# Требования к ПО

- **Требования к ПО** – описание функциональных возможностей, свойств, характеристик и ограничений ПО
- **Разработка требований** – процесс формирования требуемых значений показателей качества ПО

# Требования к ПО. Свойства

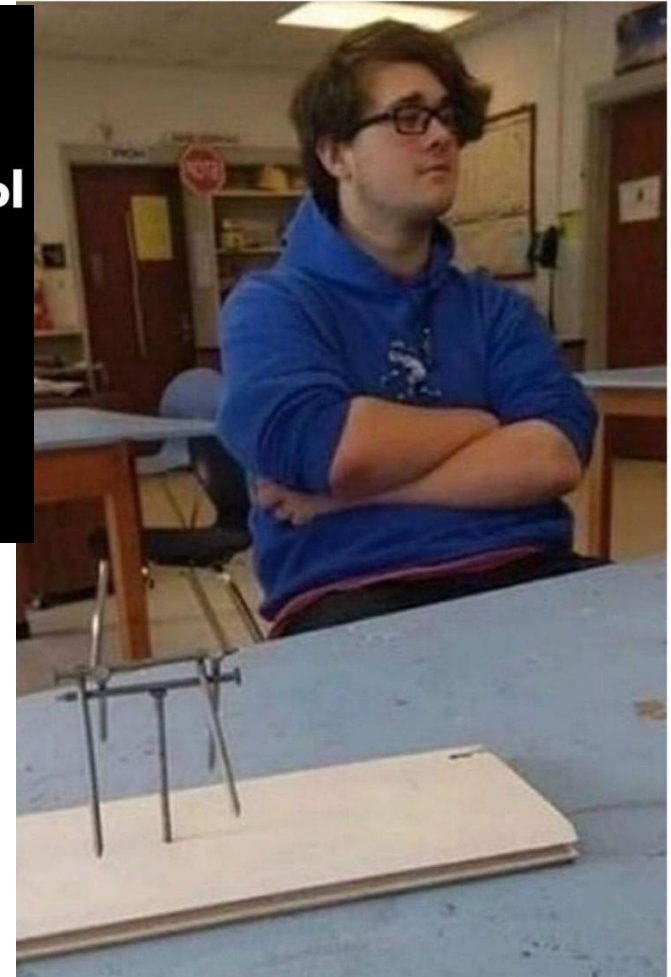
- **полнота** – охватывают все показатели качества
- **однозначность** – непротиворечивы, ясны и понятны
- **адекватность** – разумные, измеримые и достижимые, соответствие реальным потребностям
- **степень подробности** формулирования должна позволять дальнейшее проектирование
- не должны содержать конкретных проектных решений
- **зависимости** между требованиями должны быть полностью **определены**
- определены **приоритеты** требований
- для противоречивых требований определены **компромиссные соотношения**

# Требования к ПО

**АРХИТЕКТОР:**

**ИНЖЕНЕР:**

**СБАЛАНСИРОВАТЬ  
6 ГВОЗДЕЙ ТАК, ЧТОБЫ  
ОНИ НЕ КАСАЛИСЬ  
ДЕРЕВЯННОЙ  
ПОДСТАВКИ**



# Типы требований к ПО

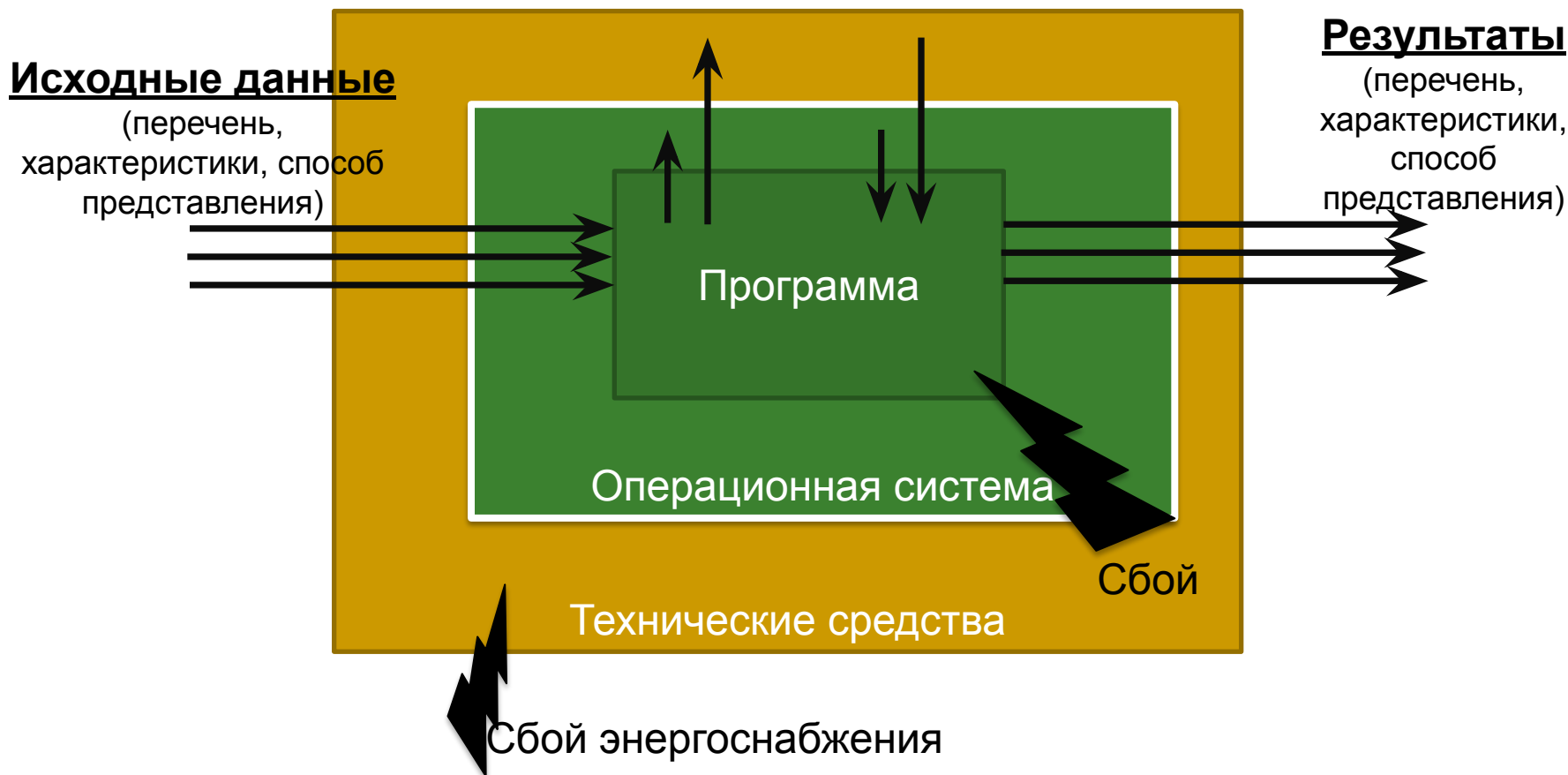
- **Функциональные** – явно описывают, что система должна делать и какие выполнять функции
  - Преобразования входных данных в выходные
- **Нефункциональные** – определяют свойства системы, напрямую не связанные с функциональностью или связанные косвенно
  - Время отклика
  - Время непрерывной работы
  - Виды документации и степень ее подробности

# Типы требований к ПО

- функциональные и технические требования – производительность, физические характеристики, окружающие условия
- к внешним интерфейсам, входным, выходным данным
- совместимость с другими системами
- безопасности
- надежности
- соответствие стандартам
- к документации
- эргономичности
- по вводу в действие и приемке – удобство развертывания
- удобство сопровождения
- удобство обучения

# Факторы, определяющие требования

## Среда функционирования



# Формирование требований

1. Анализ предметной области
2. Сбор требований
  - интервьюирование
  - построение сценариев, вариантов использования
  - наблюдение – погружение в среду
3. Классификация требований
4. Разрешение противоречий
5. Назначение приоритетов
6. Проверка требований (полнота, последовательность, непротиворечивость)
  - обзор требований для нахождения неточных описаний и ошибок
  - прототипы ПО – бета-версии, «бумажный» вариант



# Взаимодействие с пользователем

- проект, управляемый пользователем – требования полностью определяет заказчик
- проект независимый от пользователя – попытка создать ПО полезное всем
- проект, контролируемый пользователем – совместная разработка требований (дискуссия)

---

# Способы задания требований

- Варианты использования  
(пользовательские истории)
- Техническое задание
- Внешняя спецификация

---

# ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению

Документ содержащий:

- Цели разработки
- Требования
- Этапы, сроки, исполнители

---

# Последовательность разработки ТЗ

- Устанавливают набор выполняемых функций, перечень и характеристики исходных данных
- Определяют перечень результатов, их характеристики и способы представления
- Уточняют среду функционирования ПО
- Регламентация действий при сбоях и другие характеристики ПО
- Указанные сведения формируют в документ «Техническое задание» по соответствующим разделам

# Разделы ТЗ

- Введение
- Основания для разработки
- Назначение разработки
- Требования к программе:
  - требования к функциональным характеристикам
  - требования к надежности
  - условия эксплуатации
  - требования к составу и параметрам технических средств
  - требования к информационной и программной совместимости
  - требования к маркировке и упаковке
  - требования к транспортированию и хранению

---

# Разделы ТЗ

- Требования к программной документации
- Технико-экономические показатели
- Стадии и этапы разработки
- Порядок контроля и приемки
- Приложения

Возможны изменения, исключения и уточнения разделов

Если требований нет, то в соответствующем разделе указывается **«Требования не предъявляются»**

# Этапы проектирования ПО

1. **Внешнее проектирование** – проектирование взаимодействия ПО с пользователем, с последующей детализацией
2. **Проектирование архитектуры** – выделение основных подсистем и порядок взаимодействия между ними
3. **Проектирование структуры** – выделение конкретных модулей и определения порядка их взаимодействия
4. **Проектирование модулей** – точное определения интерфейса модуля, алгоритмов его функционирования, внешнее проектирование модуля, проектирование его структуры

# Внешнее проектирование

- **Внешнее проектирование** – описание поведения разрабатываемого ПО с точки зрения внешнего наблюдателя без уточнения способов реализации
- **Результат – внешняя спецификация ПО**, предназначенная для:
  - будущих пользователей (для проверки и одобрения)
  - авторов документации
  - участников всех этапов проектирования
  - тестировщиков
  - сопровождающего персонала



# Внешние спецификации

- содержат точную и полную информацию, необходимую **разработчику для построения** этого ПО, и ничего больше
- содержат точную и полную информацию, необходимую **пользователю для эксплуатации** этого ПО, и ничего больше
- позволяют проводить анализ степени соответствия этим спецификациям ПО на всех этапах проектирования
- ясность, легкость восприятия, модифицируемость
- соответствие ПО внешним спецификациям = соответствие ПО требованиям пользователей

# Внешняя спецификация модуля

- **Описание функции** – описание функций должно быть по возможности кратким и точным, без указания контекста, в котором их применение целесообразно, а также того, как эти функции могут быть реализованы
- **Обращение (порядок вызова)** – формат команд командного языка операционной системы или задания, предназначенных для вызова программы, передачи ей параметров и исходных данных, а также приема результата
- **Входные/выходные данные** – описание назначения и формата входных/выходных файлов, параметров программы
- **Внешние эффекты** – поведение программы, реакция программы на ошибки

# Методы проектирования

- **Структурное функциональное проектирование** – декомпозиция основной функции программы на функциональные подсистемы (подфункции) и далее ...
- **Структурное проектирование, основанные на потоках данных** – система = процесс преобразования входных данных в выходные
- **Объектно-ориентированное проектирование** – система = набор объектов и связей между ними

# Структурное проектирование

- Декомпозиция системы на функциональные подсистемы, затем на подфункции, затем на задачи и т.д. При этом система сохраняет целостное представление, в котором все составляющие взаимосвязаны
  - **разделение сложных задач** на множество несложных
  - **иерархическое упорядочивание** – составные части задачи организуются в древовидные структуры с добавлением новых деталей на каждом уровне
  - **абстрагирование** – выделение существенных аспектов
  - **формализация** описания задач, данных и т.п.
  - **структурирование данных** – структуризация и иерархическое представление данных
  - **непротиворечивость** – согласованность элементов

# Методология SADT

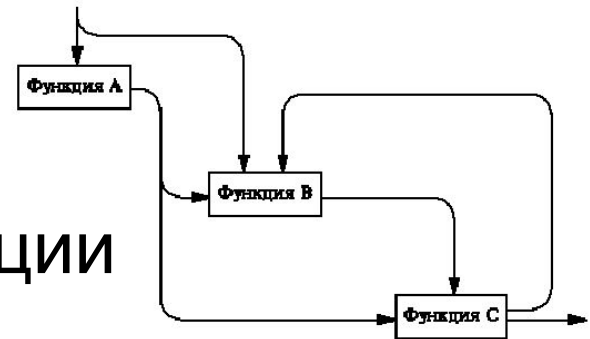
- **SADT** = Structured Analysis and Design Technique – структурный анализ и проектирование. (стандарт IDEF0)

## **SADT:**

- совокупность методов, правил и процедур, предназначенных для построения **функциональной модели системы**
- отображает функциональную структуру системы, то есть производимые ей действия и связи между действиями

# SADT

- **Диаграммы** – главные компоненты модели, все функции системы и интерфейсы на них представлены как блоки и дуги, соответственно



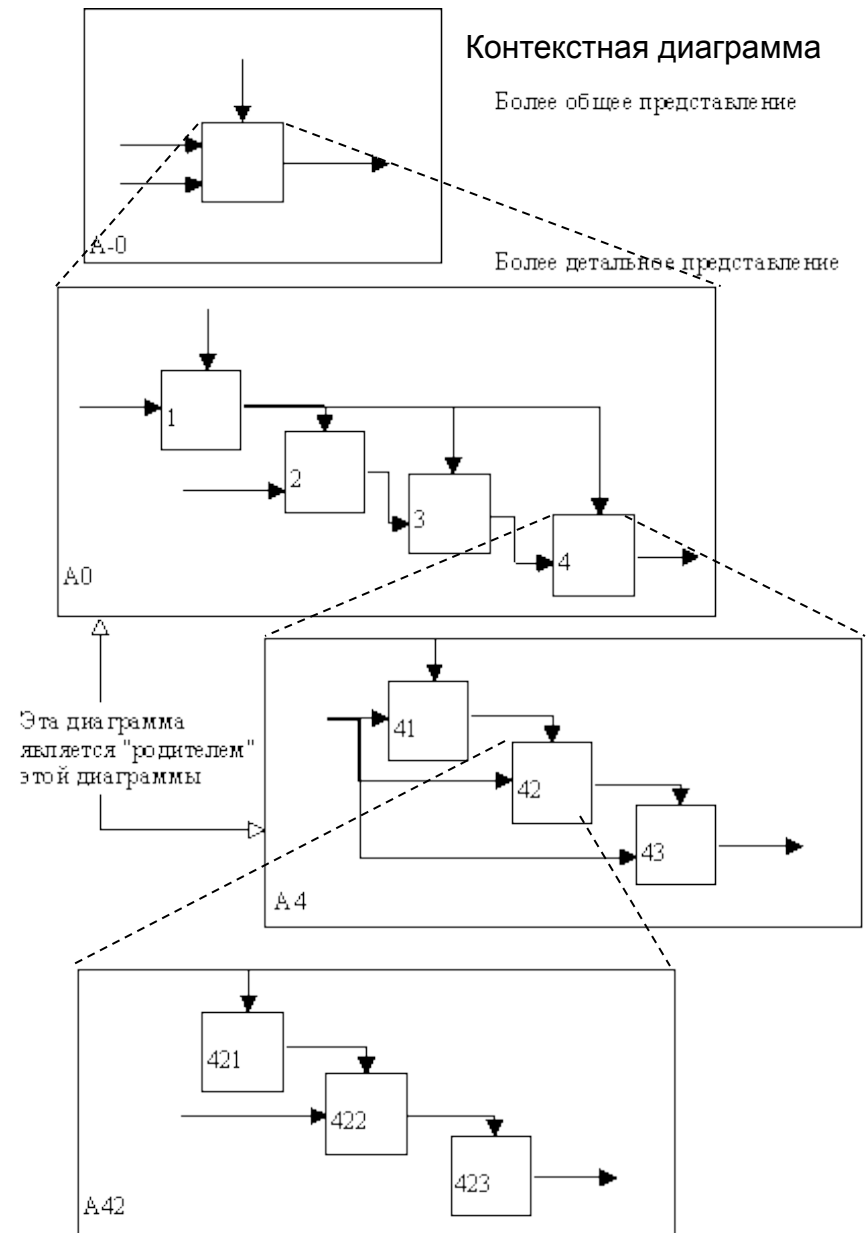
Формат описания блока





# Иерархия SADT

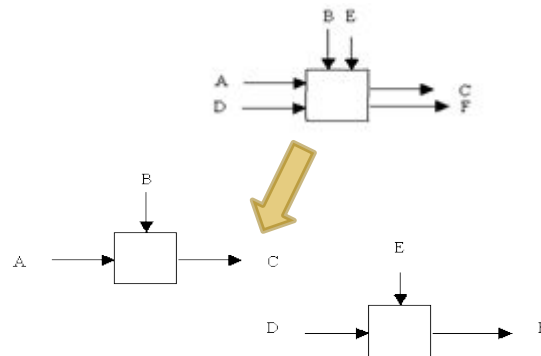
- Постепенное введение всё больших уровней детализации по мере создания диаграмм, отображающих модель
- Каждый элемент может быть подвержен декомпозиции на другой диаграмме



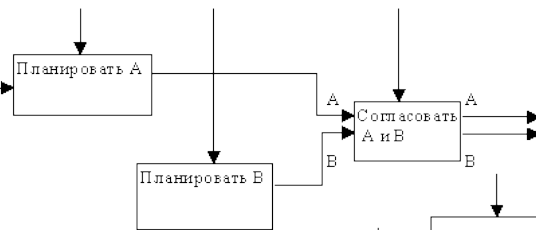


# SADT. Типы связей

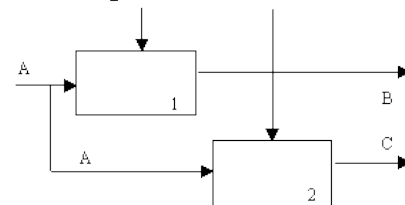
- случайная связность
- логическая связность
- временная связность (одновременно)



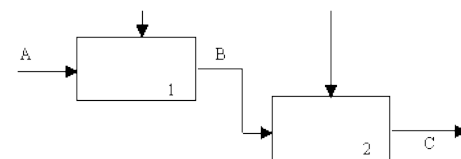
- процедурная связность



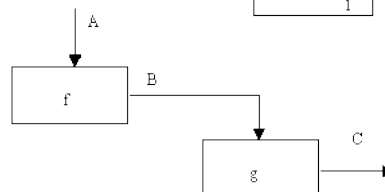
- коммуникационная связность



- последовательная связность



- функциональная связность



Значимость

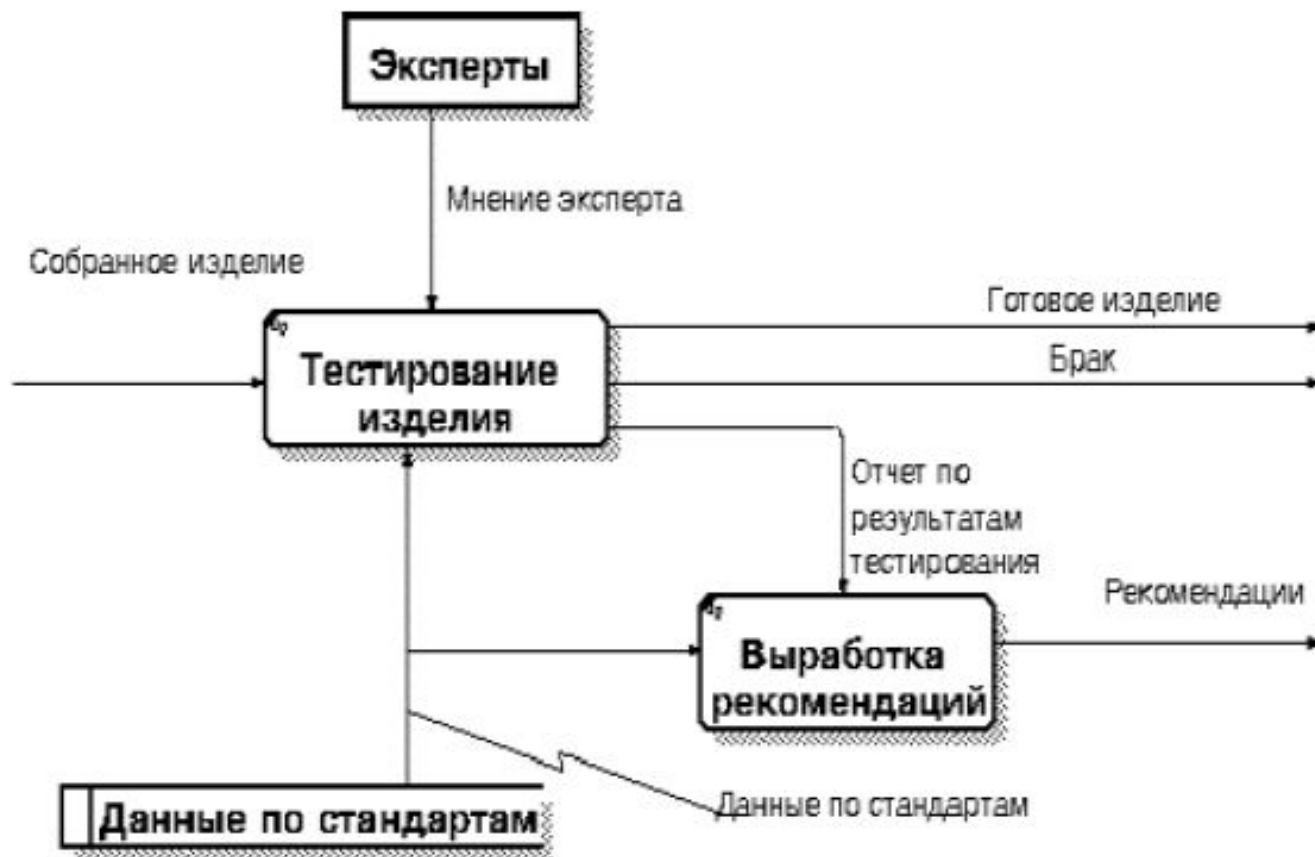
# Принципы и правила SADT

- Графическое представление системы
- Строгость и точность представления
- Ограничение количества блоков на каждом уровне декомпозиции
- Иерархическая связность диаграмм
- Уникальность меток и наименований
- Синтаксические правила для графики (блоков и дуг)
- Разделение входных данных и управляющих воздействий
- Отделение организации от функции, т.е. исключение влияния организационной структуры на функциональную модель

# Методология DFD

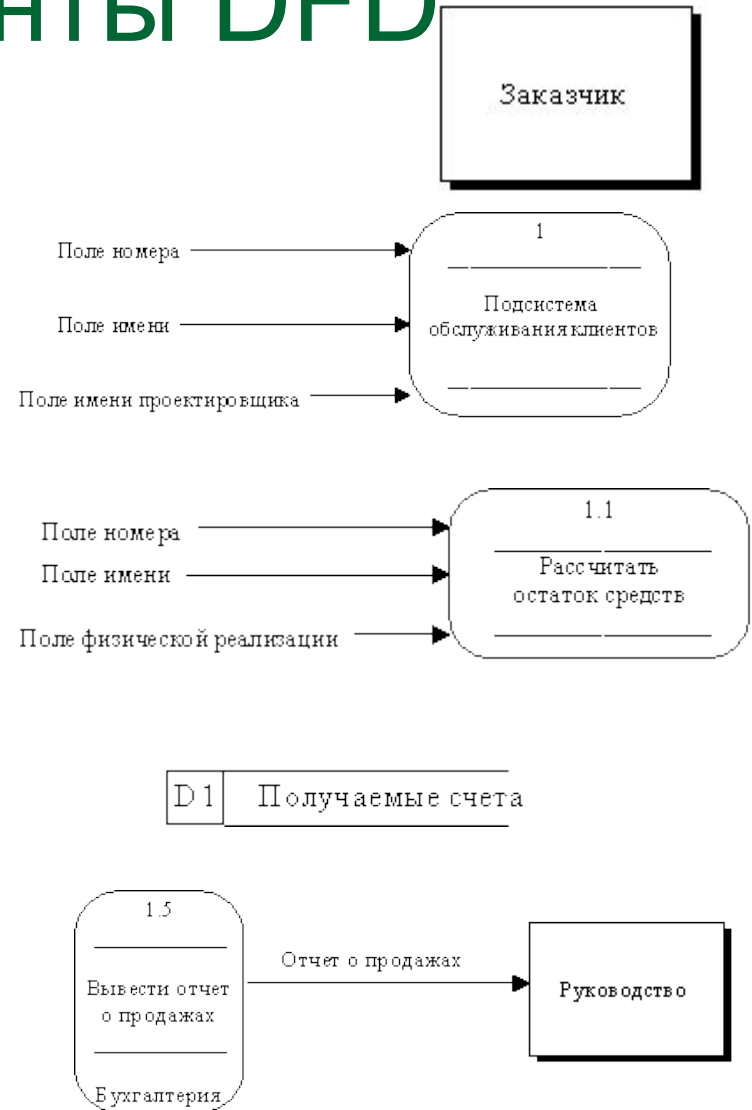
- **DFD** = Data Flow Diagram
- Модель системы – иерархия диаграмм потоков данных, описывающих асинхронный процесс преобразования информации от ее ввода в систему до выдачи пользователю
- Декомпозиция диаграмм повышает уровень детализации
- Отсутствуют понятие вход, выход, механизм, управление: блоки – процессы преобразования или хранения информации, дуги – потоки информации или данные

# DFD



# Основные компоненты DFD

- Внешние сущности (источники или потребители информации)
- Системы/подсистемы (могут быть декомпозированы)
- Процессы (преобразование входных потоков данных в выходные в соответствии с алгоритмом)
- Накопители данных (хранение информации)
- Потоки данных (информация передаваемая от источника приемнику)



# Построение SADT / DFD

1. Построение контекстных диаграмм
2. Проверка полученной модели на полноту исходных данных об объектах системы и изолированность объектов
3. Детализация подсистем и процессов
  - правило балансировки – все связи учтены, нет новых связей
  - правило иерархической нумерации
  - детализация систем прекращается если нет подсистем
  - детализация процессов прекращается если:
    - небольшое количество входных/выходных потоков данных
    - процесс может быть описан логической функцией или коротким последовательным алгоритмом

# Построение SADT / DFD

## 4. Проверка модели системы на полноту и согласованность

- В полной модели все ее объекты должны быть подробно описаны и детализированы
- В согласованной модели для всех потоков данных и объектов должно выполняться правило сохранения информации: все поступающие куда-либо данные должны быть **считаны**, а все считываемые данные должны быть **записаны**

## 5. Выявленные недетализированные объекты следует детализировать, вернувшись на предыдущие шаги

# Архитектура и структура ПО

- **Архитектура** – совокупность подсистем, образующих систему, а также порядок их взаимодействия между собой и окружающей средой
- **Построение архитектуры** – разбиение ПО на части и определение порядка взаимодействия частей
- **Цель** – упрощение процесса проектирования сложного ПО
  - раскрывает устройство системы, скрывая детали реализации
  - охватывает все варианты использования и сценарии взаимодействия
  - отвечает всем предъявляемым функциональным требованиям
  - обеспечивает требуемые показатели качества

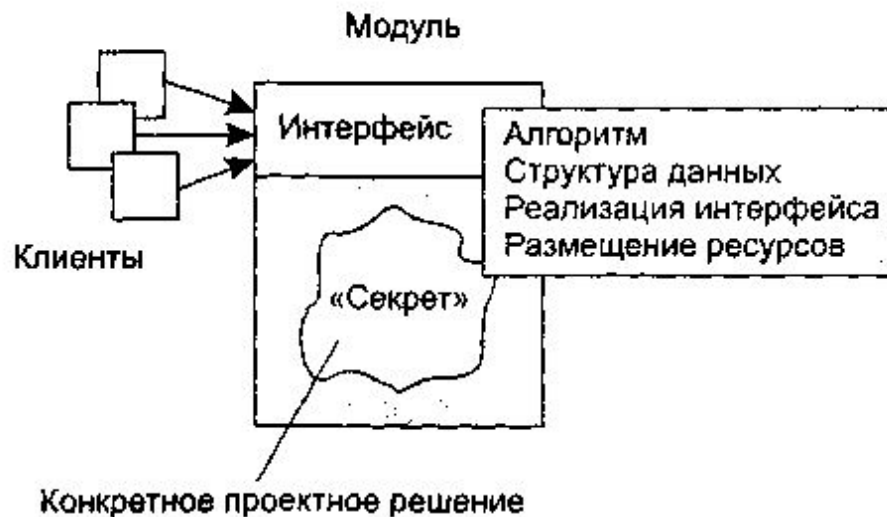


# Архитектура и структура ПО

- **Структура** – организация связи между подсистемами, а также состав и функционал подсистем
- **Проектирование структуры** – процесс детализации архитектуры:
  - определение перечня модулей, входящих в подсистемы
  - определение интерфейсов модулей
  - описание функций модулей
  - определение порядка взаимодействия модулей: связь по данным (формат данных) и управлению (порядок вызова функций)
- **Модуль** – относительно замкнутая часть программы, которая может быть вызвана из другой части программы и допускает отдельную компиляцию

# Качественная архитектура и структура

- Связь внутри подсистем (модулей) сильнее связи между подсистемами (модулями)
- Подсистема (модуль) знает о другой подсистеме (модуле) только ее внешний интерфейс



# Разбиение на модули

Снижение сложности при разбиении на модули:

- взаимодействие модулей должно быть много проще их внутренней логики
- модуль должен быть проще снаружи, чем внутренняя реализация функции
- модуль должен быть построен так, чтобы его было проще использовать повторно, чем написать заново

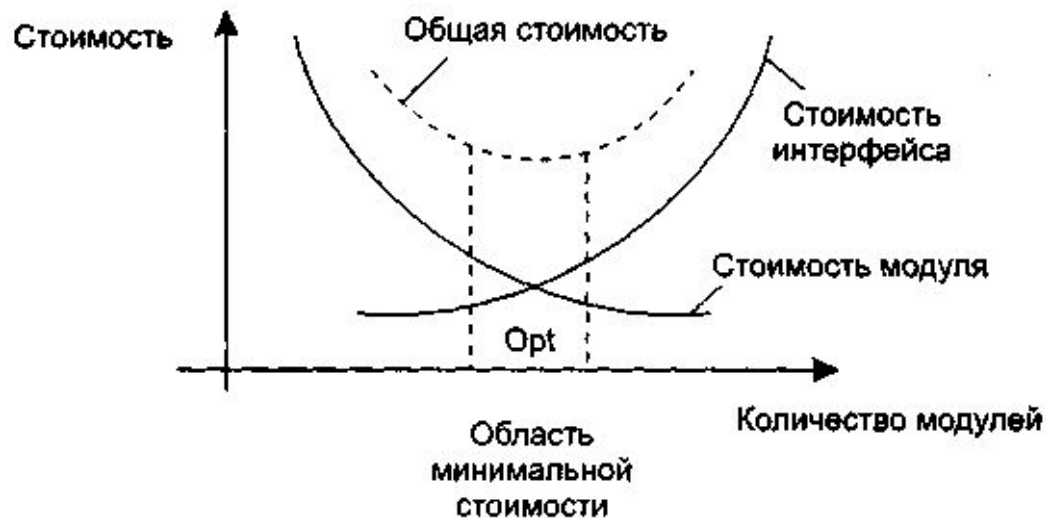
# Предел модульности

- $x$  – задача
- $C(x)$  – функция сложности решения задачи  $x$
- $T(x)$  – время решения задачи  $x$

$$C(x1) > C(x2) \quad \rightarrow \quad T(x1) > T(x2)$$

$$C(x1+x2) > C(x1)+C(x2) \quad \rightarrow \quad T(x1+x2) > T(x1)+T(x2)$$

- Сложность модулей снижается при разбиении, но растет сложность их взаимодействия



---

# Прочность модуля

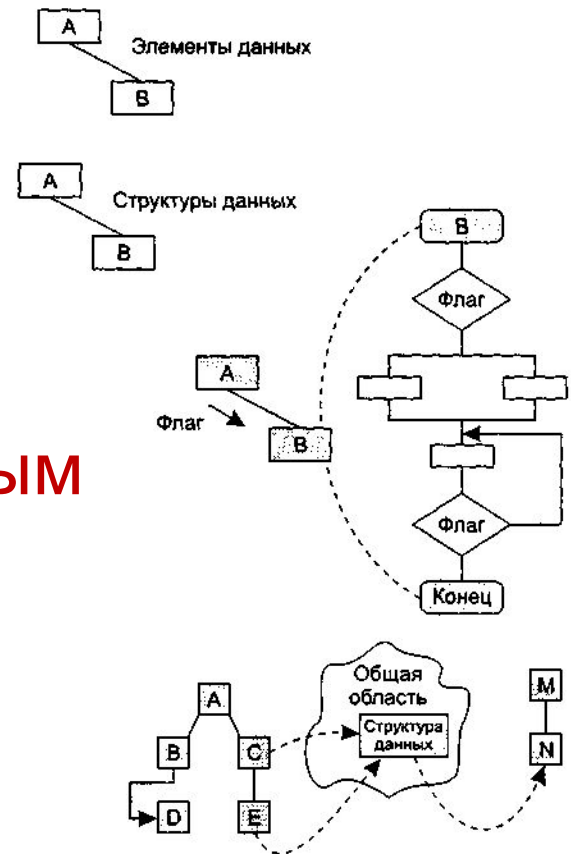
**Прочность модуля** – мера внутренних связей между функциями модуля

- Функционально прочный
- Коммуникационно-прочный
- Процедурно-прочный
- Прочный по классу
- Прочный по времени
- Прочный по логике
- Прочный по совпадению

# Сцепление модулей

– мера взаимозависимости модулей по данным  
Характеризуется способом передачи данных между модулями и свойствами самих передаваемых данных

- Сцепление по данным
- Сцепление по формату
- Сцепление по управлению
- Сцепление по внешним данным
- Сцепление по общей области
- Сцепление по содержимому



# Характеристики качества А и С

- **Размеры модулей**
- **Количество модулей**
- **Предсказуемость модулей** – информация о функционировании модулей неизменна с момента предыдущих вызовов
- **Структура принятия решения в модулях** – модули, оказывающие влияние на принятие решения подчинены модулям принимающим решения
- **Минимизация доступа к данным** – минимальное знание модуля о внешнем мире
- **Использование внутренних процедур** – процедуры, вызываемые только внутри модуля

---

# Проектирование ПО

---

Лекция 7

Тема 2: Планирование и проектирование ПО



# Вопросы

1. Требования к ПО. Функциональные и нефункциональные требования. Формирование требований. Способы извлечения требований.
2. Содержание документа «техническое задание» на разработку ПО соответствии с ГОСТ 19.201.
3. Методы внешнего проектирования ПО. Методология структурного анализа и проектирования (SADT), Моделирование потоков данных (DFD). Построение моделей и критерии завершения проектирования.
4. Этапы проектирования ПО. Внешнее проектирование ПО. Методы внешнего проектирования. Внешняя спецификация ПО и модуля.

■ ...

# Вопросы

5. Этапы проектирования ПО. Проектирование архитектуры и структуры ПО. Архитектура и структура ПО. Признаки качественной архитектуры и структуры.
6. Этапы проектирования ПО. Проектирование модулей ПО. Модуль. Прочность модуля, виды прочностей. Сцепление модулей, виды сцеплений.