

# Лекция 1: Нотация IDEF

# IDEF0 (Integration Definition for Function Modeling)

- Нотация IDEF0 была разработана на основе методологии структурного анализа и проектирования SADT, утверждена в качестве стандарта США и успешно эксплуатируется во многих проектах, связанных с описанием деятельности предприятий.




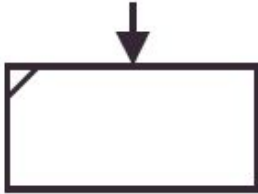
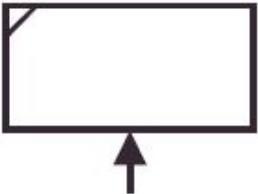
# Нотация IDEF0




- Все ресурсы, участвующие в процессе, изображаются стрелками (потоками).  
Значение стрелок различно в зависимости от того, с какой стороны процесса они расположены. Названия стрелок уникальны и формируются по правилам стандарта IDEF как обороты существительного

# Состав диаграмм IDEF0



- В нотации IDEF0 каждая процедура должна обязательно иметь не менее одной стрелки сверху, отражающей управляющее воздействие.

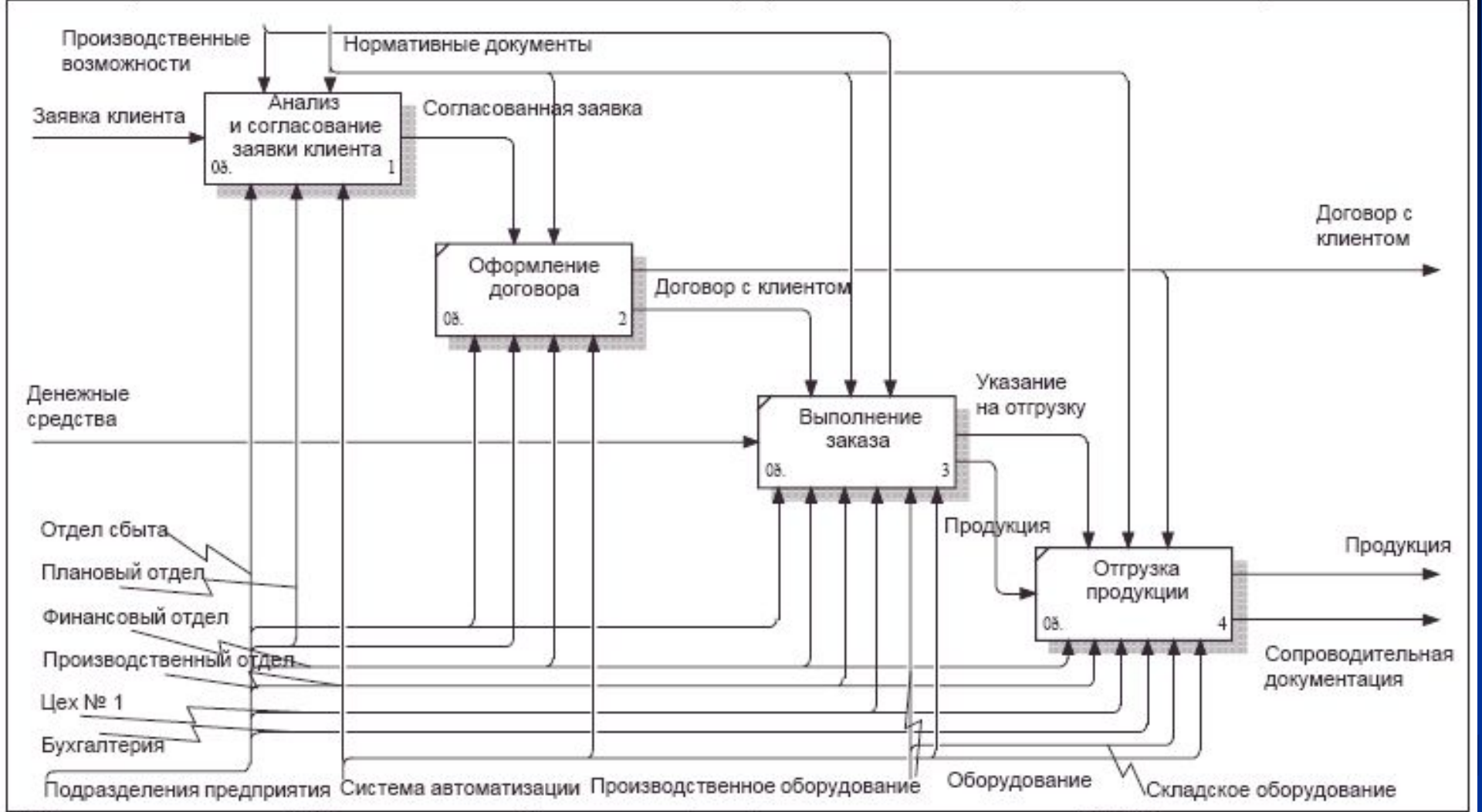
№	Наименование	Описание	Графическое представление
Нотация IDEF0			
1	Модуль поведения (UOB)	Объект служит для описания функций (процедур, работ), выполняемых подразделениями/сотрудниками предприятия.	
2	Стрелка слева	Стрелка описывает входящие документы, информацию, материальные ресурсы, необходимые для выполнения функции.	
3	Стрелка справа	Стрелка описывает исходящие документы, информацию, материальные ресурсы, являющиеся результатом выполнения функции.	
4	Стрелка сверху	Стрелка описывает управляющее воздействие, например распоряжение, нормативный документ и т.д. В нотации IDEF0 каждая процедура должна обязательно иметь не менее одной стрелки сверху, отражающей управляющее воздействие.	
5	Стрелка снизу	Стрелка снизу описывает т.н. механизмы, т.е. ресурсы, необходимые для выполнения процедуры, но не изменяющие в процессе ее выполнения свое состояние. Примеры: сотрудник, станок и т.д.	

№	Тип стрелки	Графическое представление
1	Стрелка предшествования. Соединяет последовательно выполняемые функции.	
2	Стрелка отношения. Используется для привязки объектов-комментариев к функциям.	
3	Стрелка потока объектов. Показывает поток объектов от одной функции к другой.	





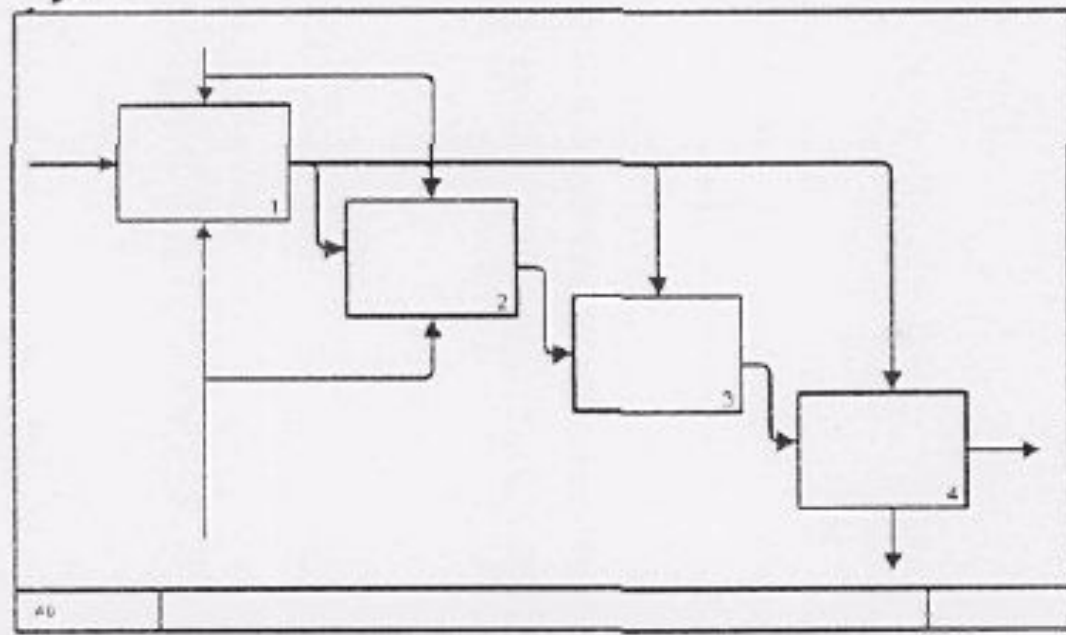
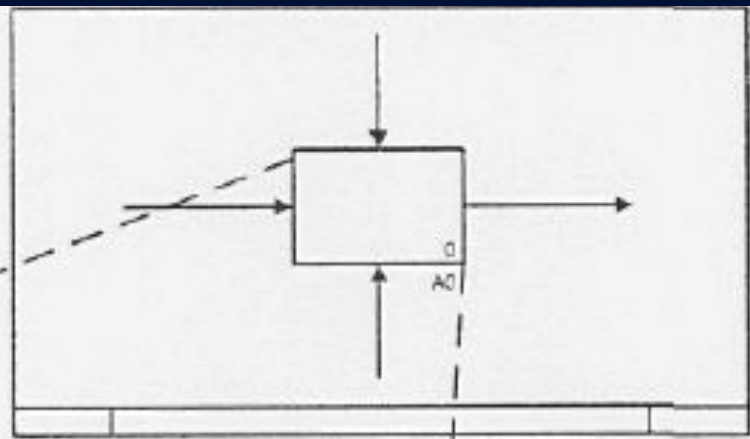
USED AT:	AUTHOR: Vladimir Repin	DATE: 30.03.01	WORKING	READER	DATE	CONTEXT:
	PROJECT: Model 1	REV: 30.03.01	DRAFT			
			RECOMMENDED			
	NOTES: 1 2 3 4 5 6 7 8 9 10		PUBLICATION			A-0



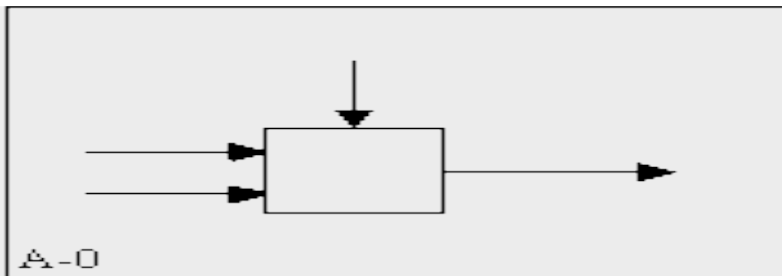
NODE:	TITLE:	NUMBER:
A0	Исполнение заказа	

# Иерархия диаграмм (декомпозиция)

Схема  
верхнего  
уровня

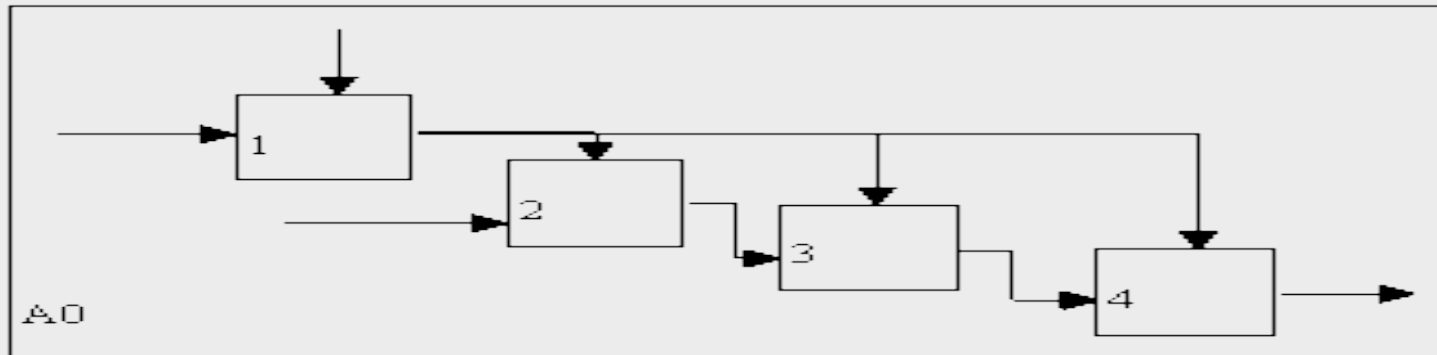


Декомпозиция

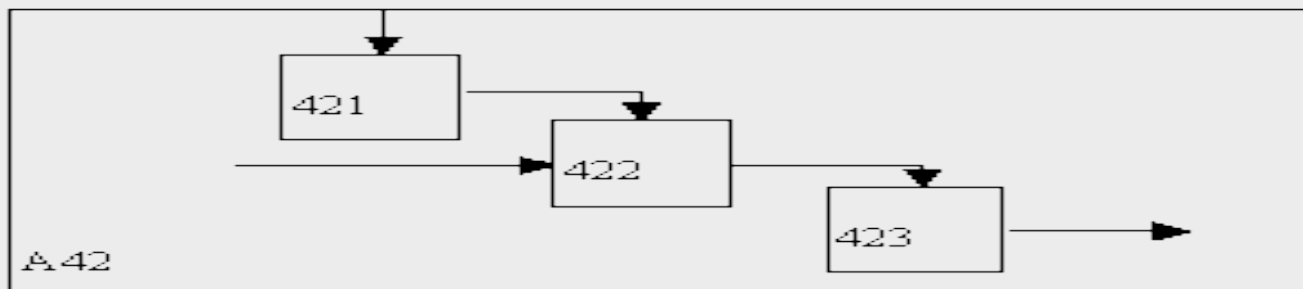
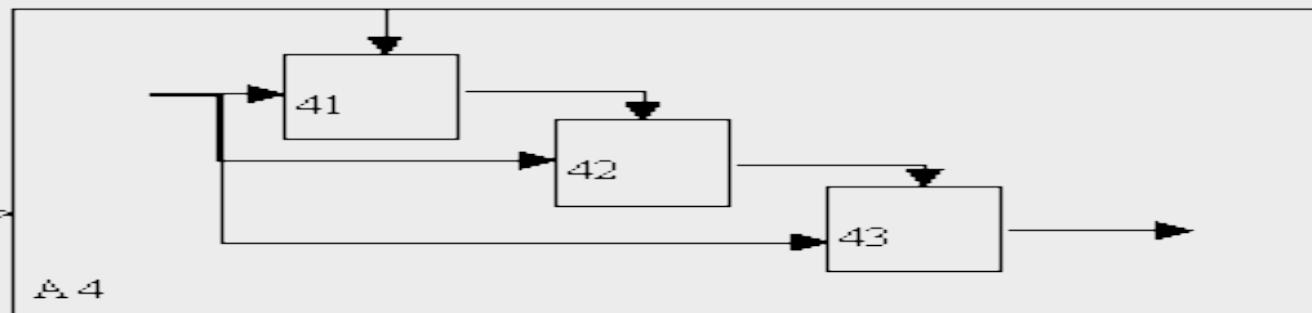


Более общее представление

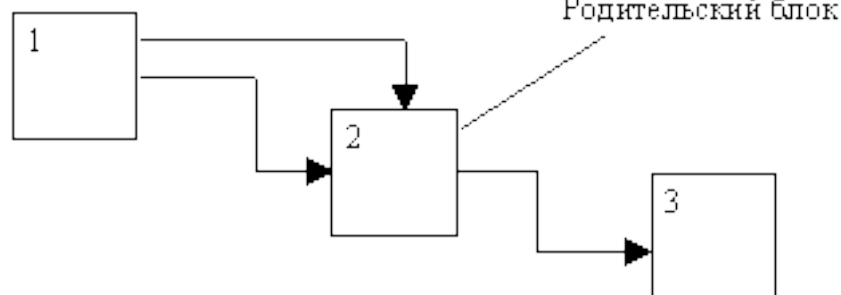
Более детальное представление



Эта диаграмма является "родителем" этой диаграммы



Родительская диаграмма



A1

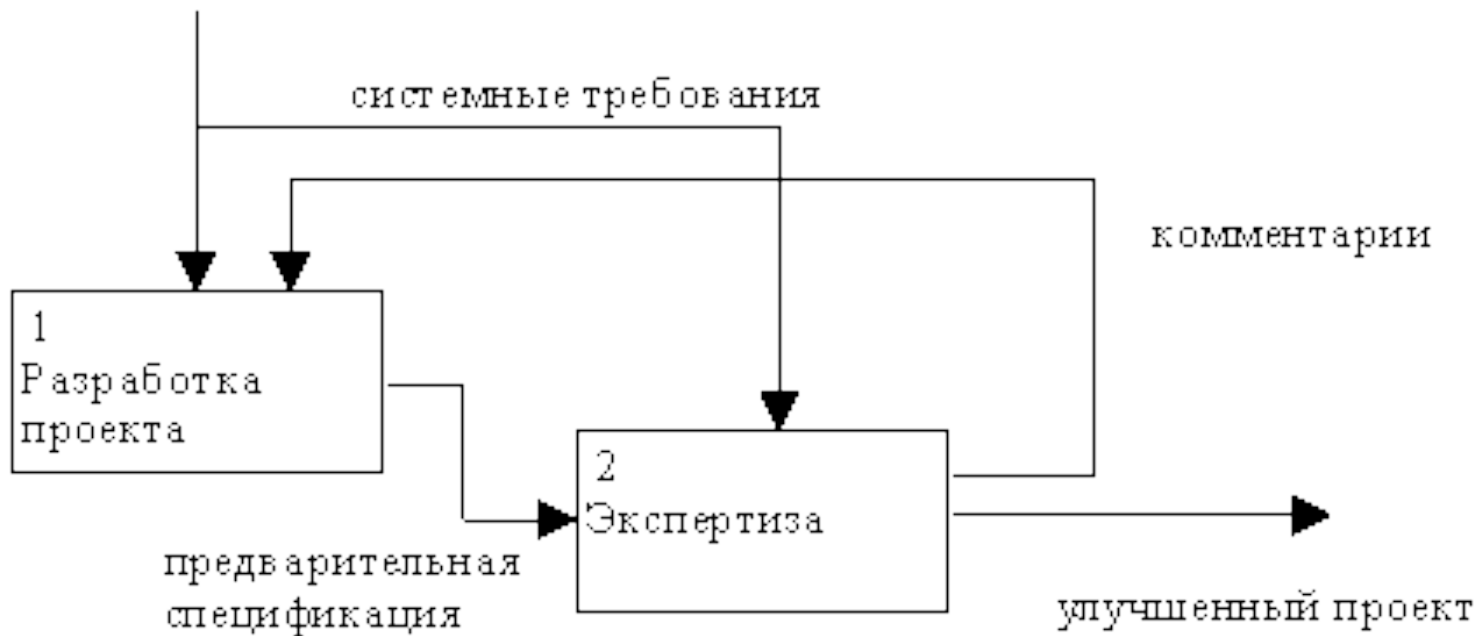
Детальная диаграмма



A12

- Неприсоединенные дуги соответствуют входам, управлениям и выходам родительского блока. Источник или получатель этих пограничных дуг может быть обнаружен только на родительской диаграмме. Неприсоединенные концы должны соответствовать дугам на исходной диаграмме. Все граничные дуги должны продолжаться на родительской диаграмме, чтобы она была полной и непротиворечивой.
- На SADT-диаграммах не указаны явно ни последовательность, ни время. Обратные связи, итерации, продолжающиеся процессы и перекрывающиеся (по времени) функции могут быть изображены с помощью дуг. Обратные связи могут выступать в виде комментариев, замечаний, исправлений и т.д.

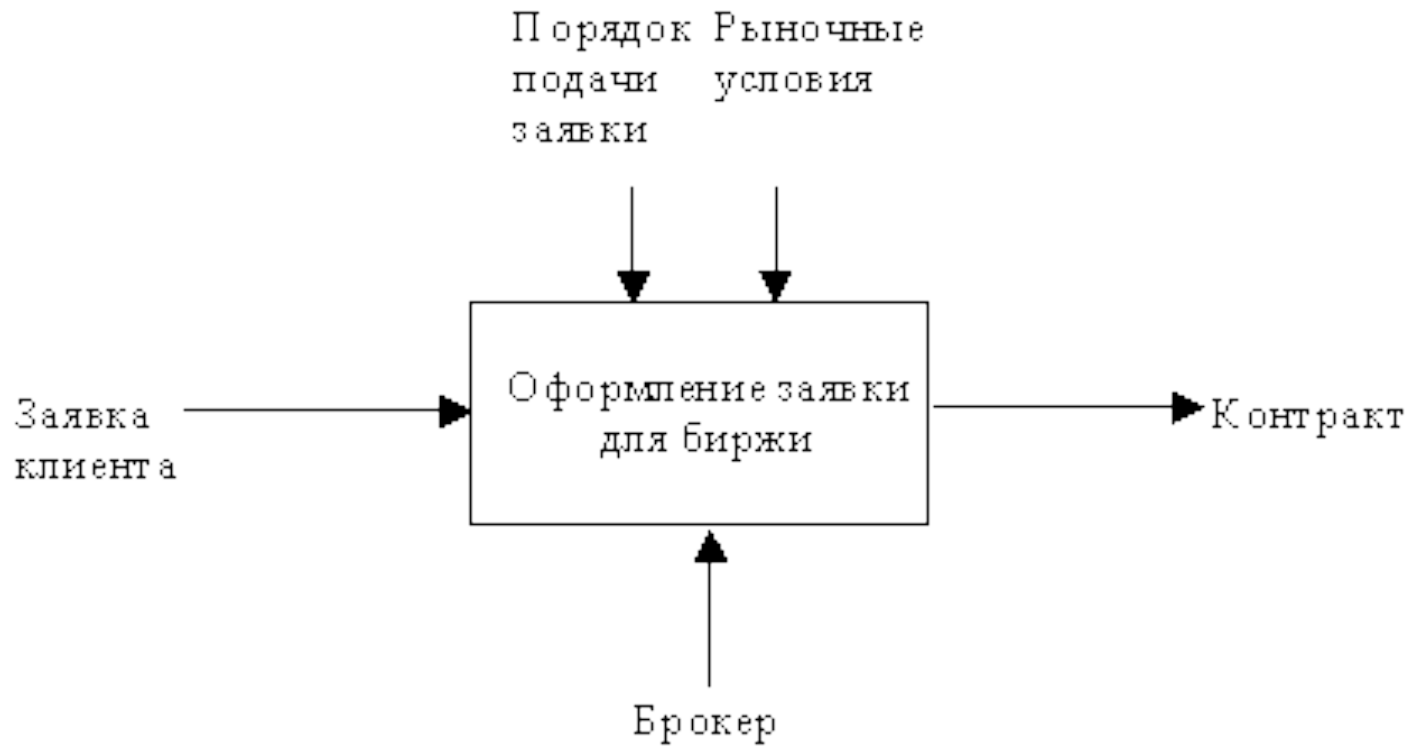
# Пример обратной связи



системные  
требования

улучшенный проект

# Пример механизма



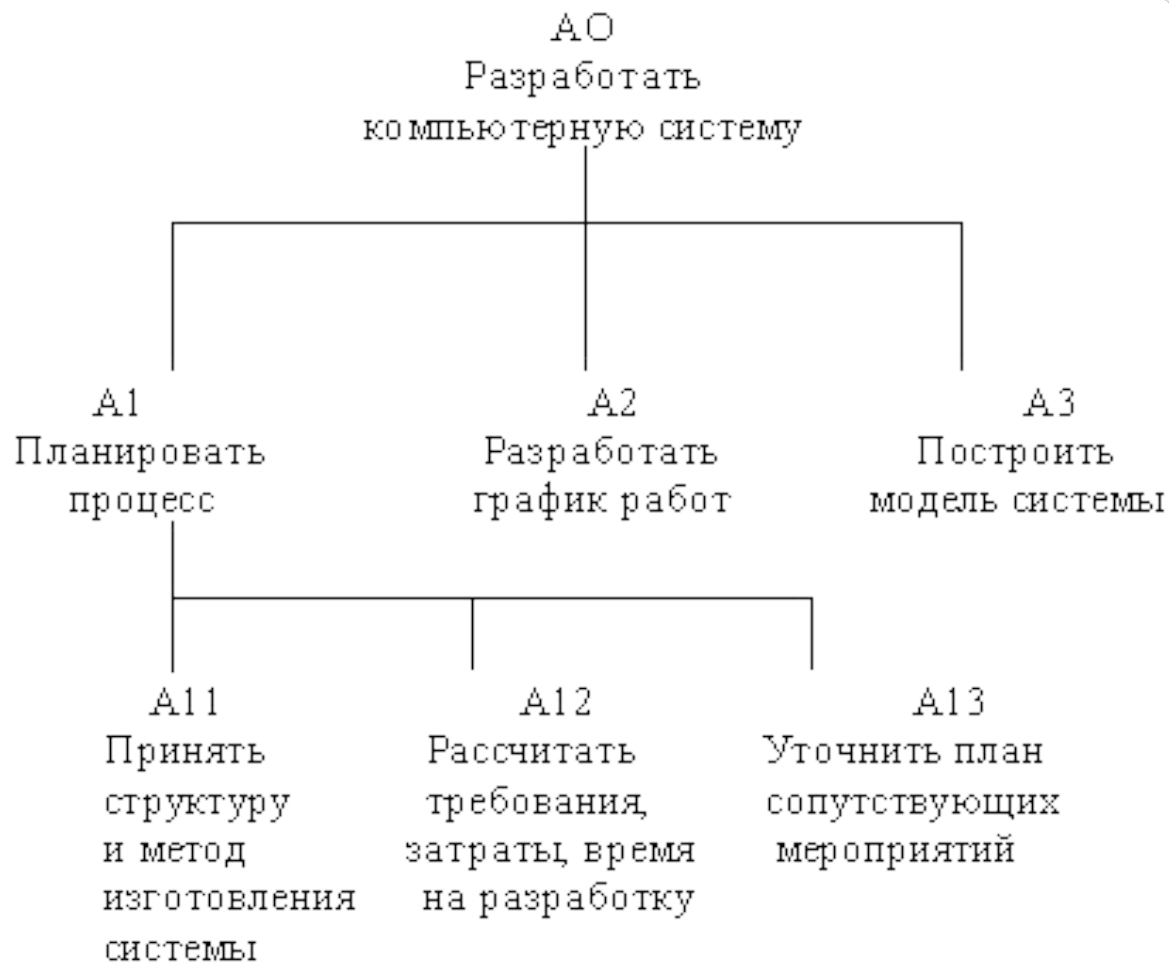
брокер



# Иерархия диаграмм

- Каждый блок на диаграмме имеет свой номер. Блок любой диаграммы может быть далее описан диаграммой нижнего уровня, которая, в свою очередь, может быть далее детализирована с помощью необходимого числа диаграмм. Таким образом, формируется иерархия диаграмм.

# Иерархия диаграмм



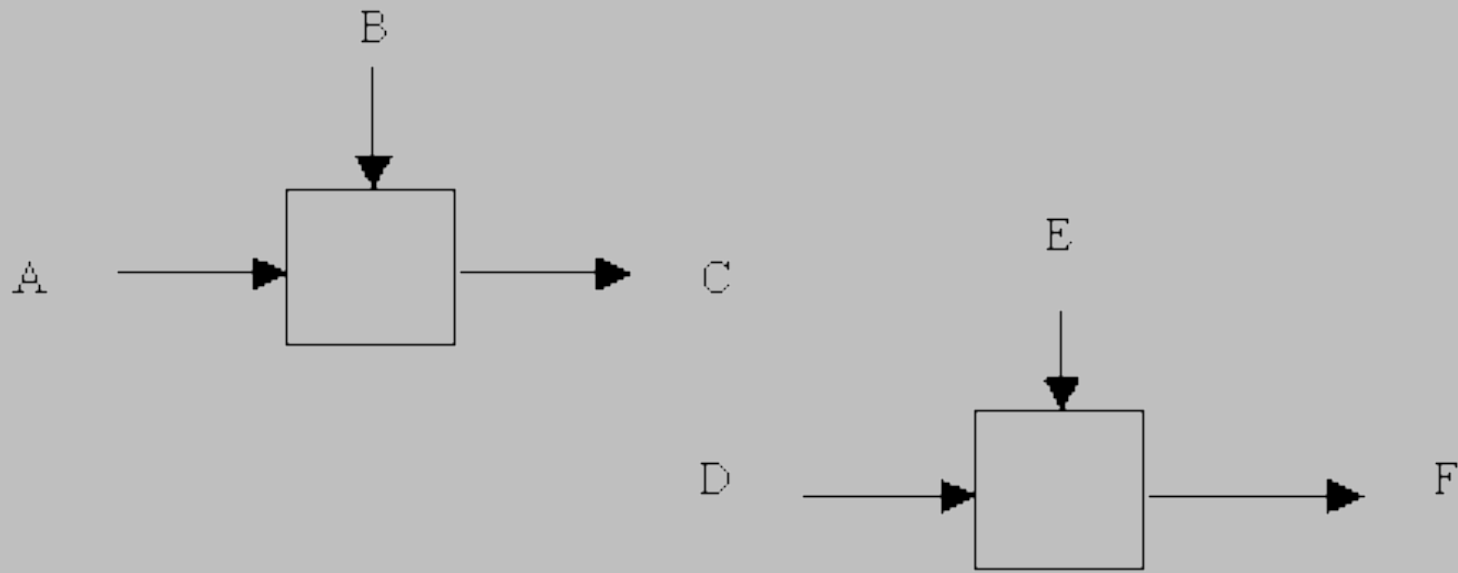
# Типы связей между функциями

- Одним из важных моментов при проектировании ИС с помощью методологии SADT является точная согласованность типов связей между функциями. Различают по крайней мере семь типов связывания:

Тип связи	Относительная значимость
Случайная	0
Логическая	1
Временная	2
Процедурная	3
Коммуникационная	4
Последовательная	5
Функциональная	6

# (0) Тип случайной связности: наименее желательный

- Случайная связность возникает, когда конкретная связь между функциями мала или полностью отсутствует



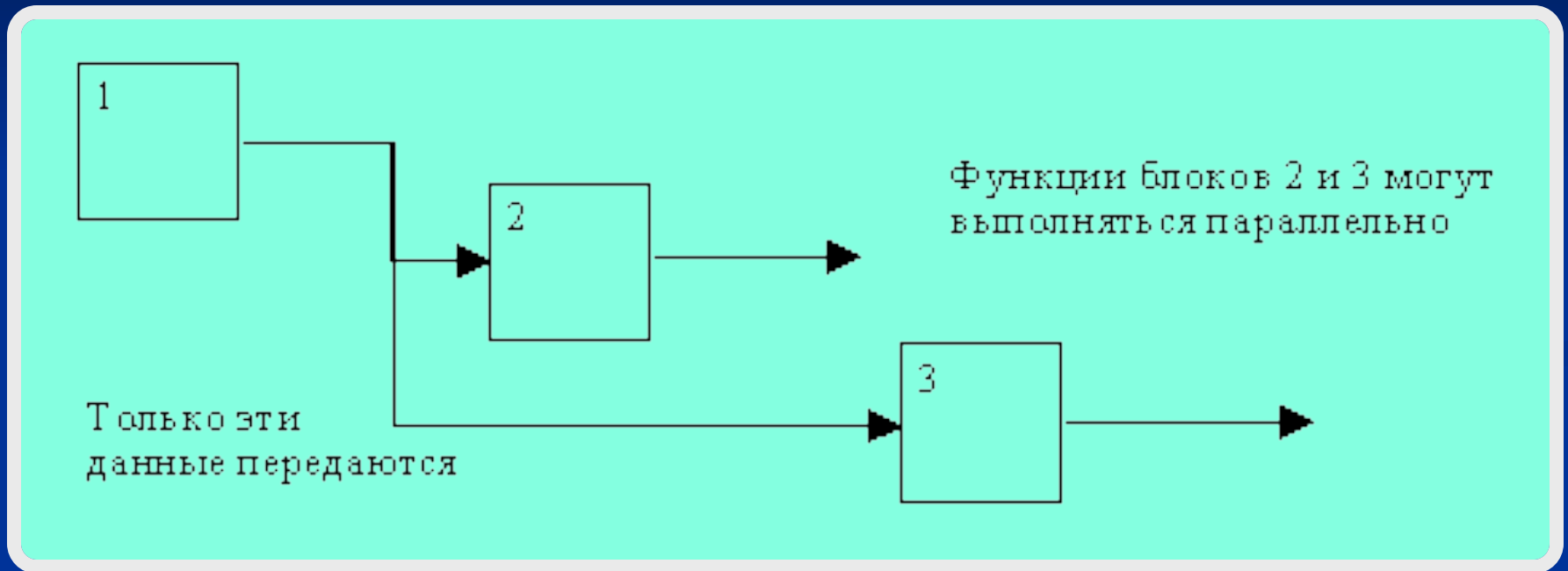
# (1) Тип логической связности

- Логическое связывание происходит тогда, когда данные и функции собираются вместе вследствие того, что они попадают в *общий класс* или *набор* элементов, но необходимых функциональных отношений между ними не обнаруживается.

## (2) Тип временной связности

- Связанные по времени элементы возникают вследствие того, что они представляют функции, связанные во времени, когда данные используются одновременно или функции включаются параллельно, а не последовательно.

# Одновременное выполнение



Только эти данные передаются

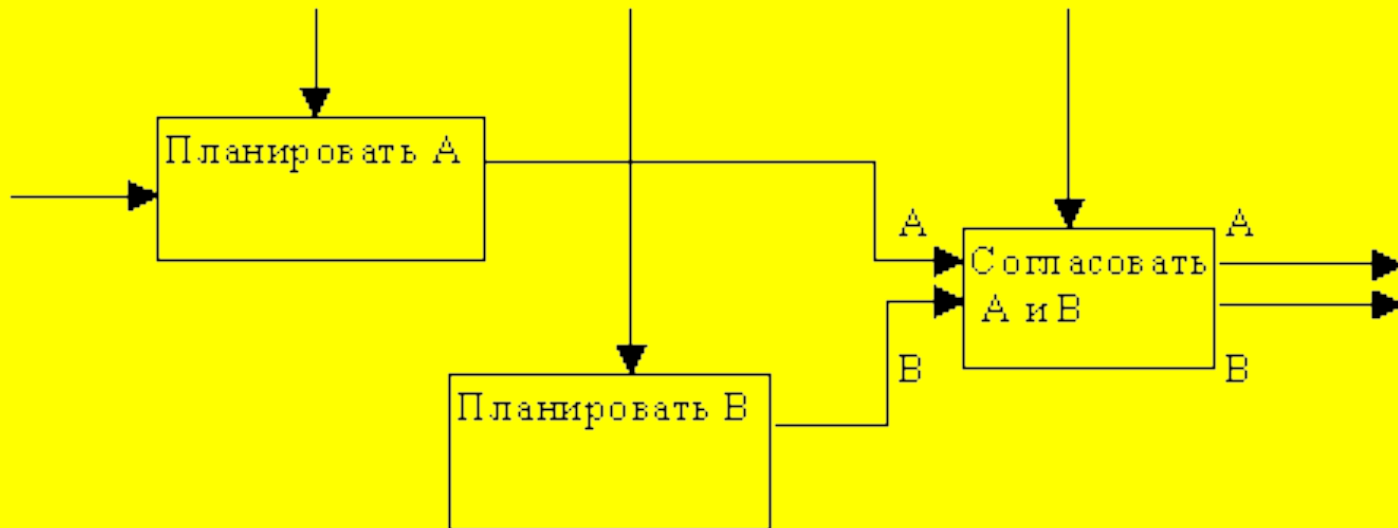


## (3) Тип процедурной связности

- Процедурно-связанные элементы появляются сгруппированными вместе вследствие того, что они выполняются в течение *одной и той же части* цикла или процесса.



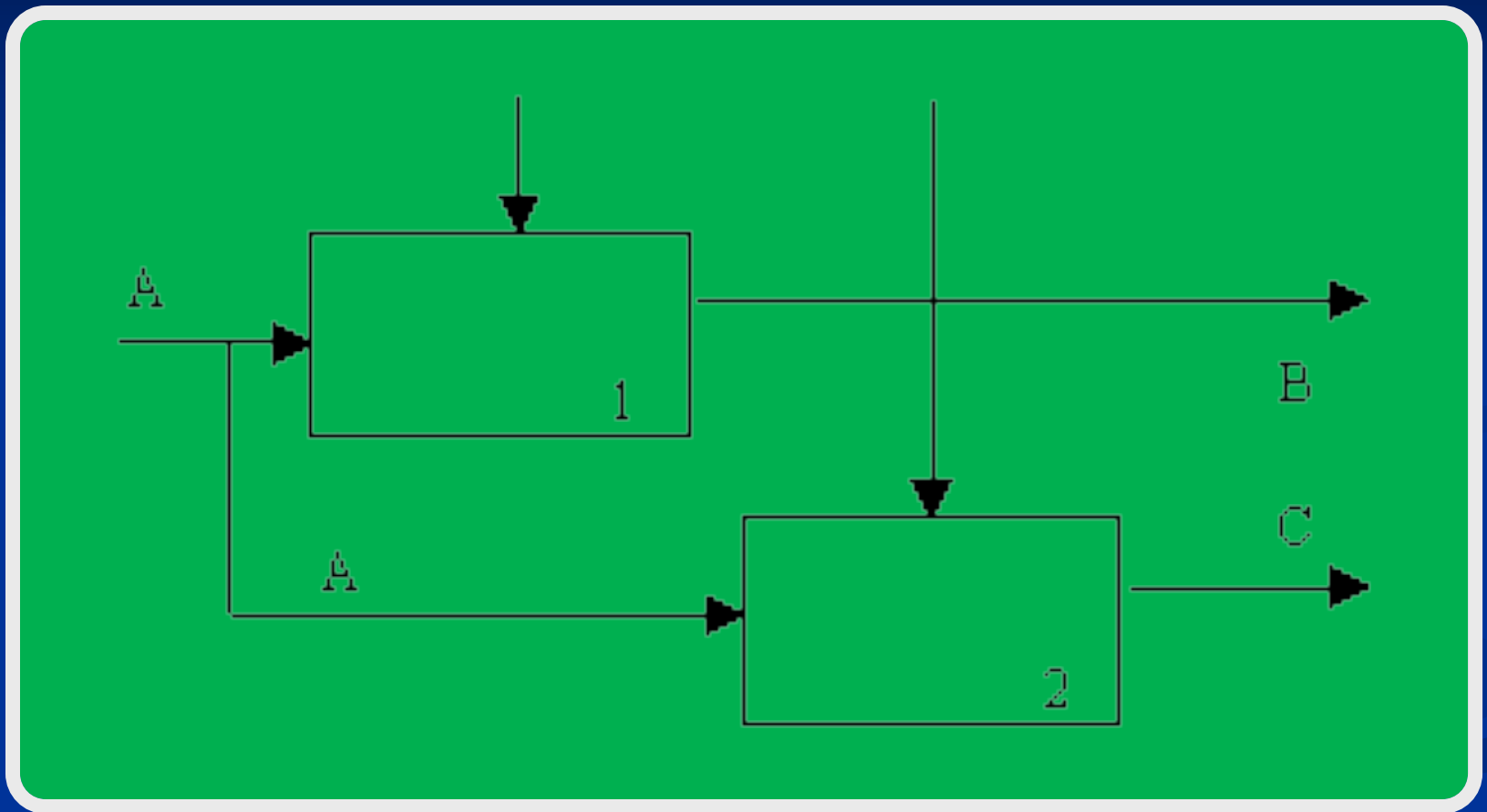
# Процедурная связность



# (4) Тип коммуникационной связности

- Диаграммы демонстрируют коммуникационные связи, когда блоки группируются вследствие того, что они используют одни и те же *входные данные* и/или производят одни и те же *выходные данные*

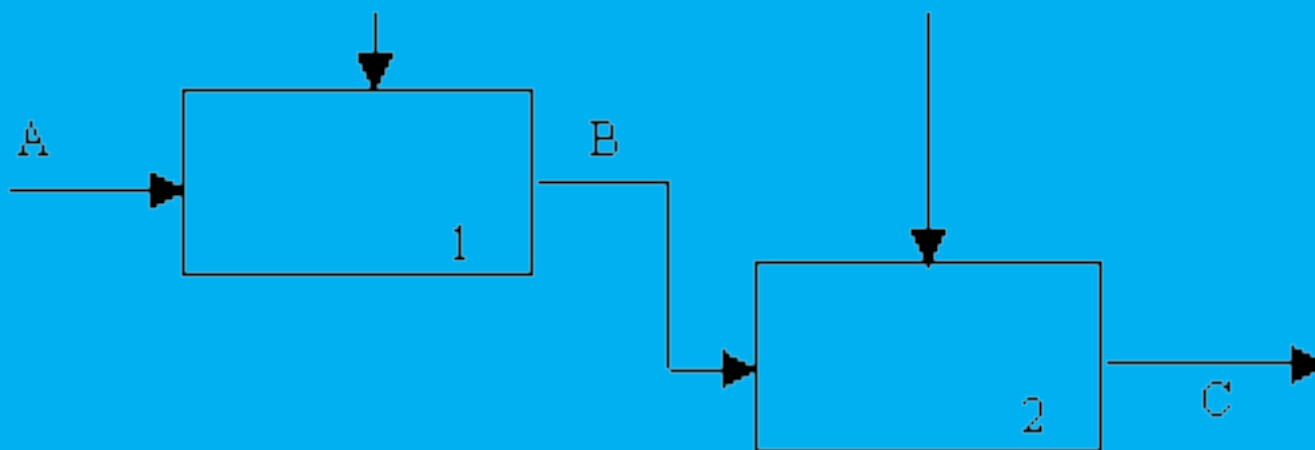
# Коммуникационная связность



## (5) Тип последовательной СВЯЗНОСТИ

- На диаграммах, имеющих последовательные связи, *выход одной функции служит входными данными для следующей функции.*
- Связь между элементами на диаграмме является более тесной, чем на рассмотренных выше уровнях связей, поскольку моделируются причинно-следственные зависимости

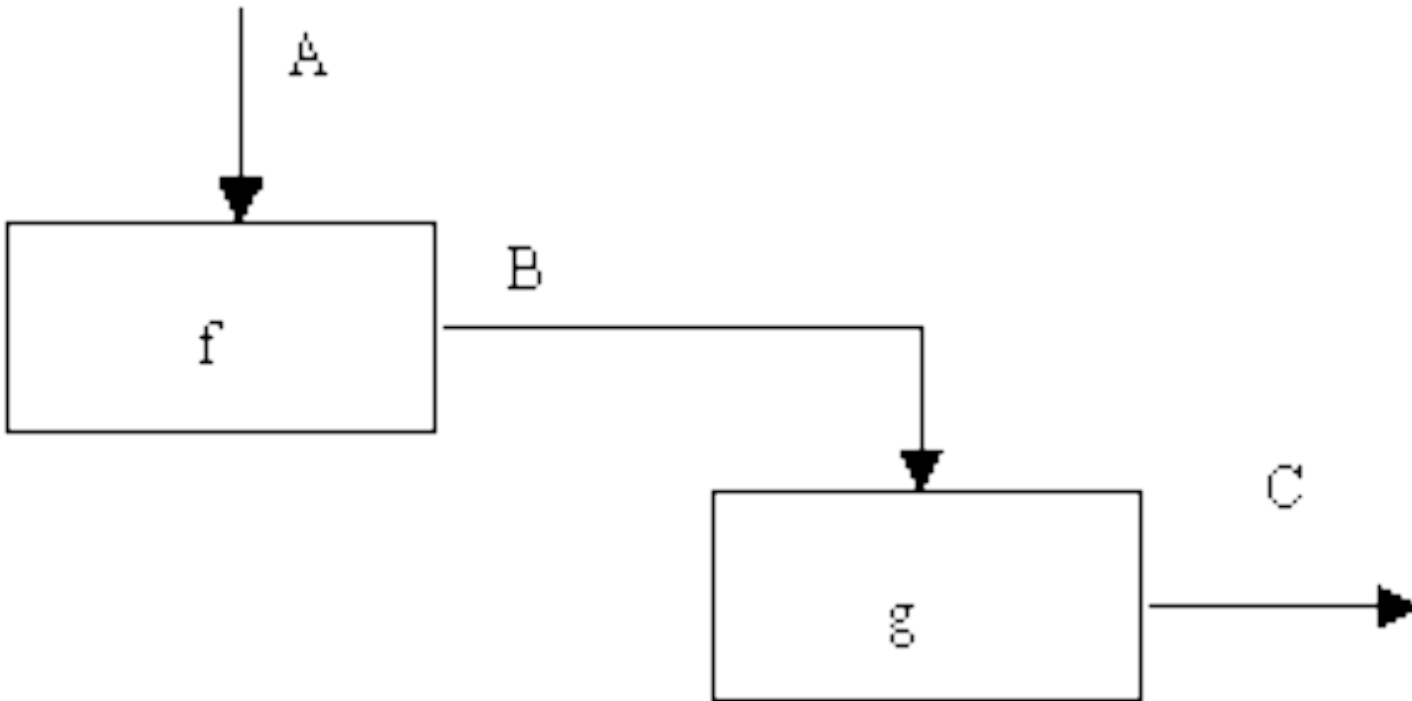
# Последовательная связность



## (6) Тип функциональной СВЯЗНОСТИ

- Диаграмма отражает полную функциональную связность, при наличии полной зависимости одной функции от другой. Диаграмма, которая является чисто функциональной, не содержит чужеродных элементов, относящихся к последовательному или более слабому типу СВЯЗНОСТИ.

# Функциональная связность



- $C = g(B) = g(f(A))$

Благодарю за внимание!