



ІІТМО

Практическая работа 6

Методология IDEF

1. Используя шаблон курсовой работы и правила оформления работ (ГОСТ 7.32), создать отчет по ПР6.
 - Титульный лист – отчет по практической работе 6
 - Задания – нет
 - Введение – пара фраз о реальном содержании работы.
 - Основная часть должна содержать краткое описание предметной области функционирования приложения или системы.
 - Для рассматриваемого приложения/ системы необходимо построить функциональную модель в стандарте IDEF0. Количество уровней декомпозиции – 3 (для одной или нескольких работ 1 уровня) .
 - Предусмотреть обратные связи между работами по входу или управлению.
 - В качестве инструмента необходимо использовать любой инструмент, поддерживающий построение моделей в стандарте IDEF0.
 - Заключение – вывод по работе.
 - Список литературы должен содержать ссылки на все источники, использованные в работе. В тексте **не забывать ссылаться** на источники из списка.
2. **Дедлайн – 10 .11**



Функциональная модель

Функциональная модель предназначена для описания существующих бизнес-процессов, в котором используются как естественный, так и графический языки.

Для передачи информации о конкретной системе источником графического языка является сама методология IDEF0.

Методология IDEF0

Методология IDEF0 предписывает построение иерархической системы диаграмм - единичных описаний фрагментов системы. ✕

Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция - система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности.



Методология IDEF0

Каждая **IDEF0-диаграмма** содержит блоки и дуги.



- Блоки изображают функции моделируемой системы, на диаграммах изображаются прямоугольниками, означаями поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Имя работы должно быть выражено отглагольным существительным, обозначающим действие.
- Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними.

Методология IDEF0

IDEF0 требует, чтобы в диаграмме было не менее трех и не более шести блоков. Эти ограничения поддерживают сложность диаграмм и модели на уровне, доступном для чтения, понимания и использования.  

Каждая сторона блока имеет вполне определенное назначение. Левая сторона блока предназначена для входов, верхняя - для управления, правая - для выходов, нижняя - для механизмов.

Такое обозначение отражает определенные системные принципы: входы преобразуются в выходы управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают, кем и чем выполняется функция.

Методология IDEF0

Блоки в IDEF0 размещаются по степени важности, как ее понимает автор диаграммы. Этот относительный порядок называется доминированием.

Доминирование понимается как влияние, которое один блок оказывает на другие блоки диаграммы. Например, самым доминирующим блоком диаграммы может быть либо первый из требуемой последовательности функций, либо планирующая или контролирующая функция, влияющая на все другие.

Наиболее доминирующий блок обычно размещается в верхнем левом углу диаграммы, а наименее доминирующий - в правом углу.

Методология IDEF0

Расположение блоков на странице отражает авторское определение доминирования.



Таким образом, топология диаграммы показывает, какие функции оказывают большее влияние на остальные. Чтобы подчеркнуть это, аналитик может перенумеровать блоки в соответствии с порядком их доминирования. Порядок доминирования может обозначаться цифрой, размещенной в правом нижнем углу каждого прямоугольника: 1 будет указывать на наибольшее доминирование, 2 - на следующее и т. д.

Взаимодействие работ с внешним миром и между собой описывается в виде стрелок, изображаемых одинарными линиями со стрелками на концах. Стрелки представляют собой некую информацию и именуются существительными.

Типы стрелок

В IDEF0 различают пять типов стрелок.

Вход - объекты, используемые и преобразуемые работой для получения результата (выхода). Допускается, что работа может не иметь ни одной стрелки входа. Стрелка входа рисуется как входящая в левую грань работы.

Управление - информация, управляющая действиями работы. Обычно управляющие стрелки несут информацию, которая указывает, что должна выполнять работа. Каждая работа должна иметь хотя бы одну стрелку управления, которая изображается как входящая в верхнюю грань работы.

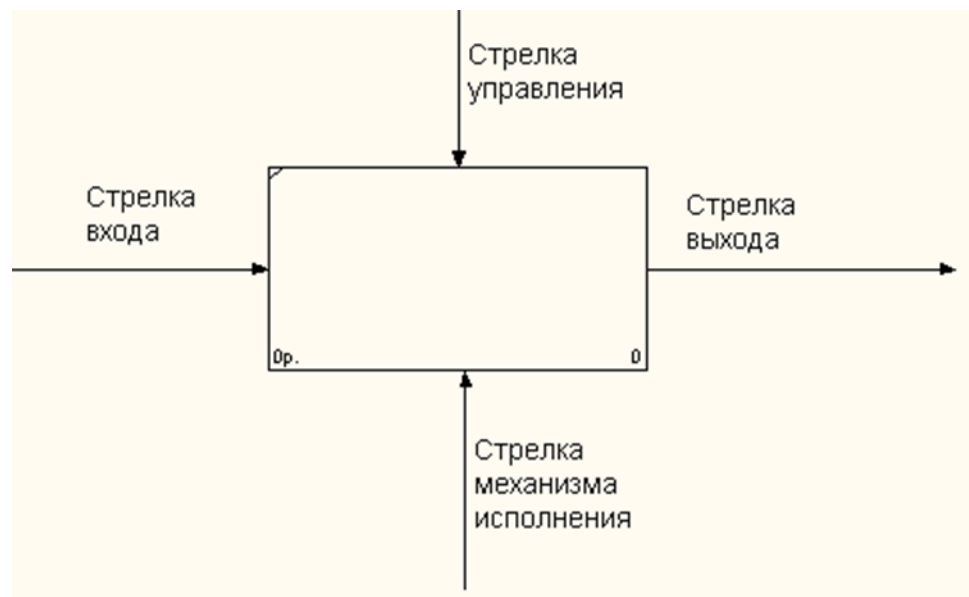
Выход - объекты, в которые преобразуются входы. Каждая работа должна иметь хотя бы одну стрелку выхода, которая рисуется как исходящая из правой грани работы.

Механизм - ресурсы, выполняющие работу. Стрелка механизма рисуется как входящая в нижнюю грань работы. По усмотрению аналитика стрелки механизма могут не изображаться на модели.

Вызов - специальная стрелка, указывающая на другую модель работы. Стрелка вызова рисуется как исходящая из нижней части работы и используется для указания того, что некоторая работа выполняется за пределами моделируемой системы.



Типы стрелок



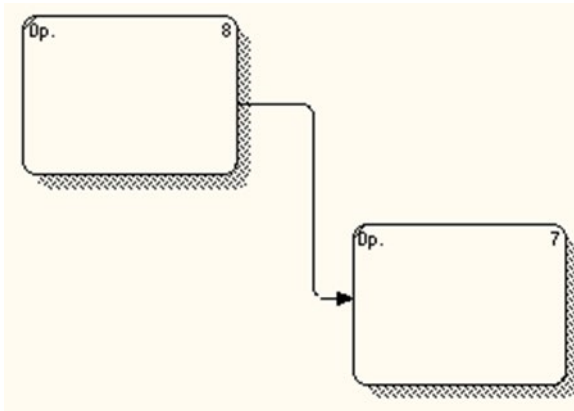
Типы взаимодействий между блоками ИТМО

В методологии IDEF0 требуется только пять типов взаимодействий между блоками для описания их отношений: управление, вход, обратная связь по управлению, обратная связь по входу, выход-механизм. Связи по управлению и входу являются простейшими, поскольку они отражают прямые воздействия, которые интуитивно понятны и очень просты.

Типы взаимодействий между блоками ИТМО



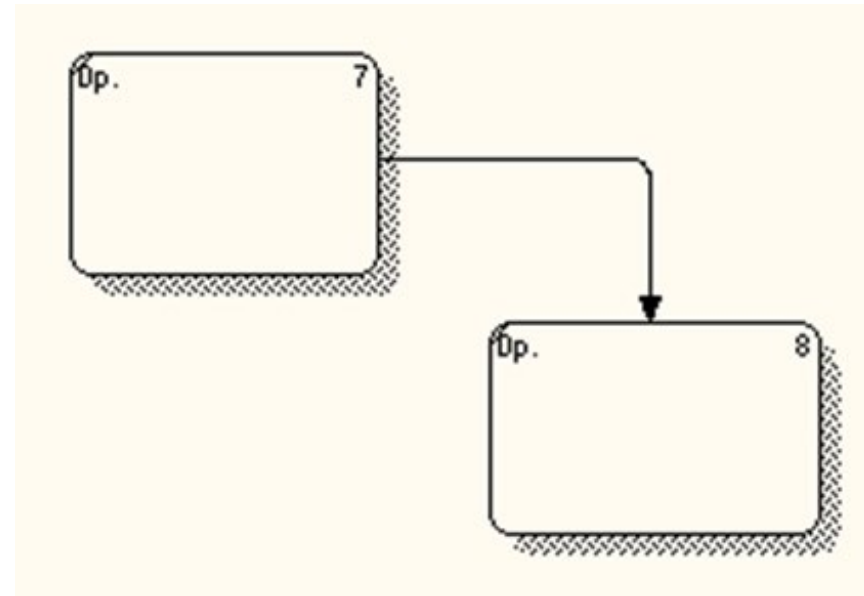
Связь по входу



Типы взаимодействий между блоками ИТМО

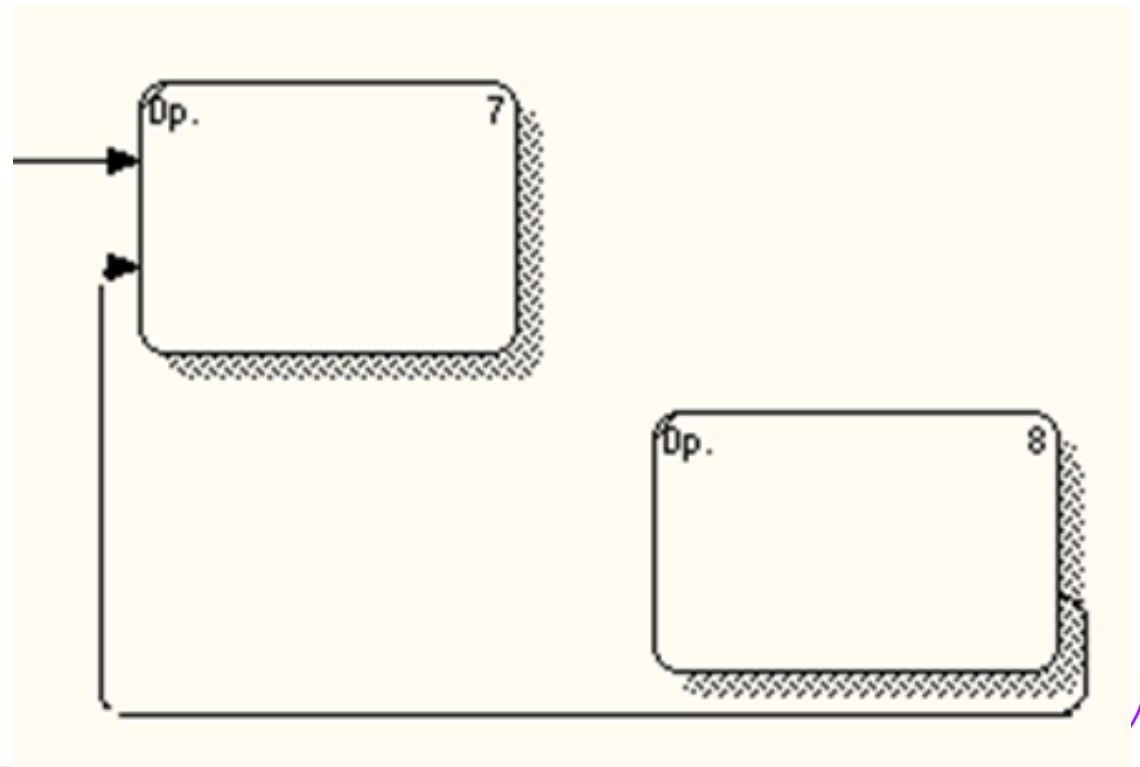
Связь по управлению

- Отношение управления возникает тогда, когда выход одного блока непосредственно влияет на блок с меньшим доминированием.



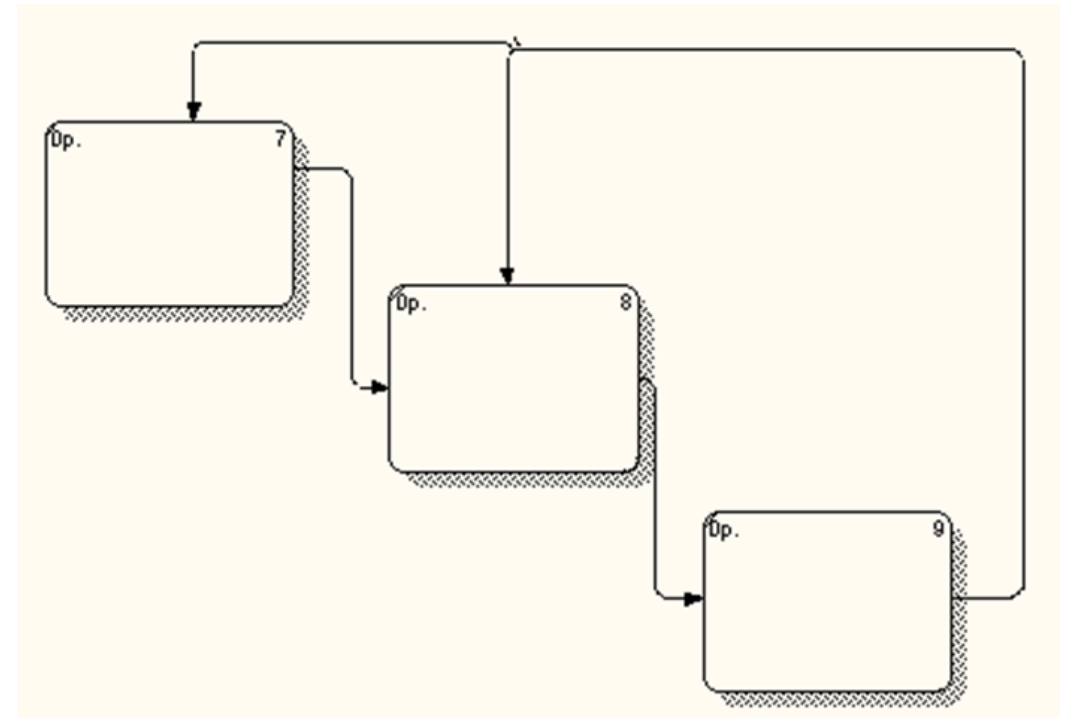
Типы взаимодействий между блоками ИТМО

Обратная связь по входу
возникает когда выходы из
одной работы влияют на
будущее выполнение других
работ, что впоследствии
повлияет на исходную работу



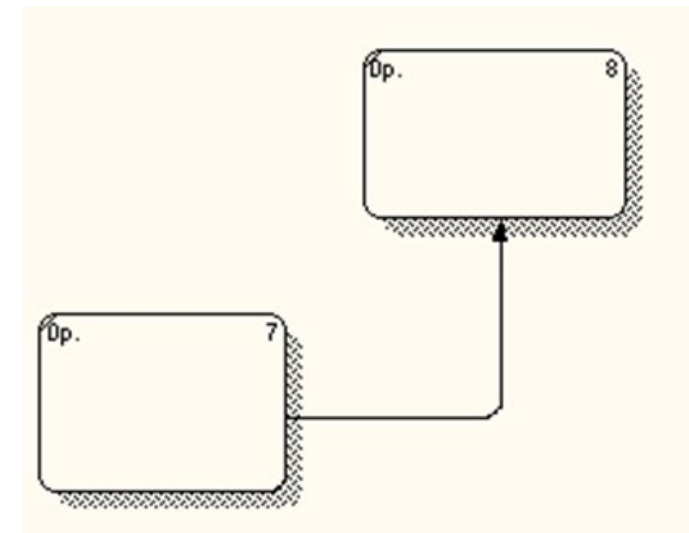
Типы взаимодействий между блоками ИТМО

Обратная связь по управлению возникает тогда когда выход некоторого блока влияет на блок с большим доминированием.




Типы взаимодействий между блоками ИТМО

Связи «выход-механизм» характерны при распределении источников ресурсов (например, требуемые инструменты, обученный персонал, физическое пространство, оборудование, финансирование, материалы).



Дуги

В IDEF0 дуга редко изображает один объект. Обычно она символизирует набор объектов. Так как дуги представляют наборы объектов, они могут иметь множество начальных точек (источников) и конечных точек (назначений). Поэтому дуги могут разветвляться и соединяться различными способами. Вся дуга или ее часть может выходить из одного или нескольких блоков и заканчиваться в одном или нескольких блоках.

Разветвление дуг, изображаемое в виде расходящихся линий, означает, что  все содержимое дуг или его часть может появиться в каждом ответвлении. Дуга всегда помечается до разветвления, чтобы дать название всему набору. Кроме того, каждая ветвь дуги может быть помечена или не помечена в соответствии со следующими правилами:

- непомеченные ветви содержат все объекты, указанные в метке дуги перед разветвлением;
- ветви, помеченные после точки разветвления, содержат все объекты или их часть, указанные в метке дуги перед разветвлением.

Слияния дуг в IDEFO, изображаемое как сходящиеся вместе линии, указывает, что содержимое каждой ветви идет на формирование метки для дуги, являющейся результатом слияния исходных дуг. После слияния результирующая дуга всегда помечается для указания нового набора объектов, возникшего после объединения. Кроме того, каждая ветвь перед слиянием может помечаться или не помечаться в соответствии со следующими правилами:

- непомеченные ветви содержат все объекты, указанные в общей метке дуги после слияния;
- помеченные перед слиянием ветви содержат все или некоторые объекты из перечисленных в общей метке после слияния.

Пример

Построение модели системы должно начинаться с изучения всех документов, описывающих ее функциональные возможности. Одним из таких документов является техническое задание, а именно разделы "Назначение разработки", "Цели и задачи системы" и "Функциональные характеристики системы".

После изучения исходных документов и опроса заказчиков и пользователей системы необходимо сформулировать цель моделирования и определить точку зрения на модель.

Рассмотрим технологию ее построения на примере системы "Служба занятости в рамках вуза".

Пример

Сформулируем цель моделирования:

- описать функционирование системы, которое было бы понятно ее пользователю, не вдаваясь в подробности, связанные с реализацией.

Модель будем строить с точки зрения пользователей (студент, преподаватель, администратор, деканат, фирма).

Начнем с построения контекстной IDEF0-диаграммы.

Основной функцией является обслуживание ее клиентов посредством обработки запросов, от них поступающих. Таким образом, определим единственную работу контекстной диаграммы как «Обслужить клиента системы». Далее определим входные и выходные данные, а также механизмы и управление.

Для того чтобы обслужить клиента, необходимо зарегистрировать его в системе, открыть доступ к БД и обработать его запрос.

В качестве входных данных будут использоваться «имя клиента», «пароль клиента», «исходная БД», «запрос клиента». Выполнение запроса ведет либо к получению информации от системы, либо к изменению содержимого БД (например, при составлении экспертных оценок), поэтому выходными данными будут являться «отчеты» и «измененная БД». Процесс обработки запросов будет выполняться монитором системы под контролем администратора.



Пример

Контекстная диаграмма



Пример

Проведем декомпозицию контекстной диаграммы, описав последовательность обслуживания клиента:

- Определение уровня доступа систему.
- Выбор подсистемы.
- Обращение к подсистеме.
- Изменение БД (при необходимости).



Пример. Декомпозиция 1 уровня ИТМО

Проведем декомпозицию контекстной диаграммы, описав последовательность обслуживания клиента:

- Определение уровня доступа в систему.
- Выбор подсистемы.
- Обращение к подсистеме.
- Изменение БД (при необходимости).



Пример. Декомпозиция блока «Определение уровня доступа в систему»

Декомпозируем последовательно все блоки полученной диаграммы.

Первым этапом при определении уровня доступа в систему является определение категории пользователя.

По имени клиента осуществляется поиск в базе пользователей, определяя его категорию. Согласно определенной категории выясняются полномочия, предоставляемые пользователю системы.

Далее проводится процедура доступа в систему, проверяя имя и пароль доступа. Объединяя информацию о полномочиях и уровне доступа в систему, для пользователя формируется набор разрешенных действий.



Пример. Блок «Обращение к подсистеме»

После прохождения процедуры доступа в систему монитор анализирует запрос клиента, выбирая подсистему, которая будет обрабатывать запрос.

Декомпозиция работы «Обращение к подсистеме» не отвечает цели и точке зрения модели.

Пользователя системы не интересуют внутренние алгоритмы ее работы. В данном случае ему важно, что выбор подсистемы будет произведен автоматически, без его вмешательства, поэтому декомпозиция обращения к подсистеме не имеет смысла.

Пример. Декомпозиция блока «Обработка запроса клиента»

Перед осуществлением поиска ответа на запрос необходимо подключиться к БД. В общем случае БД может находиться на удаленном сервере, поэтому может потребоваться установление соединения с ней.

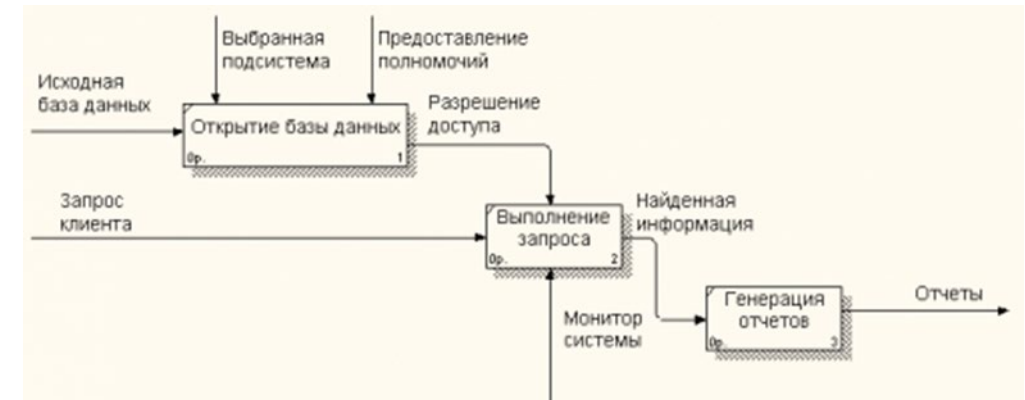
Определим последовательность работ:

- Открытие БД.
- Выполнение запроса.
- Генерация отчетов.

После открытия БД необходимо сообщить системе об установлении соединения с БД, после чего выполнить запрос и сгенерировать отчеты для пользователя.

Необходимо отметить, что в «Выполнение запроса» включается работа различных подсистем.

- Например, если запрос включает в себя тестирование, то его будет исполнять подсистема профессиональных и психологических тестов. На этапе выполнения запроса может потребоваться изменение содержимого БД, например при составлении экспертных оценок. Поэтому, на диаграмме необходимо предусмотреть такую возможность.



Корректировка диаграммы

При анализе полученной диаграммы возникает вопрос, по каким правилам происходит генерация отчетов?



Необходимо наличие заранее сформированных шаблонов, по которым будет производиться выборка из БД, причем эти шаблоны должны соответствовать запросам и должны быть заранее определены. Кроме того, клиенту должна быть предоставлена возможность выбора формы отчета.

Скорректируем диаграмму, добавив в нее стрелки «Шаблоны отчетов» и «Запросы на изменение БД» и туннельную стрелку «Клиент системы». Туннелирование «Клиента системы» применено для того, чтобы не выносить стрелку на диаграмму верхнего, так как функция выбора формы отчета не является достаточно важной для отображения ее на родительской диаграмме.

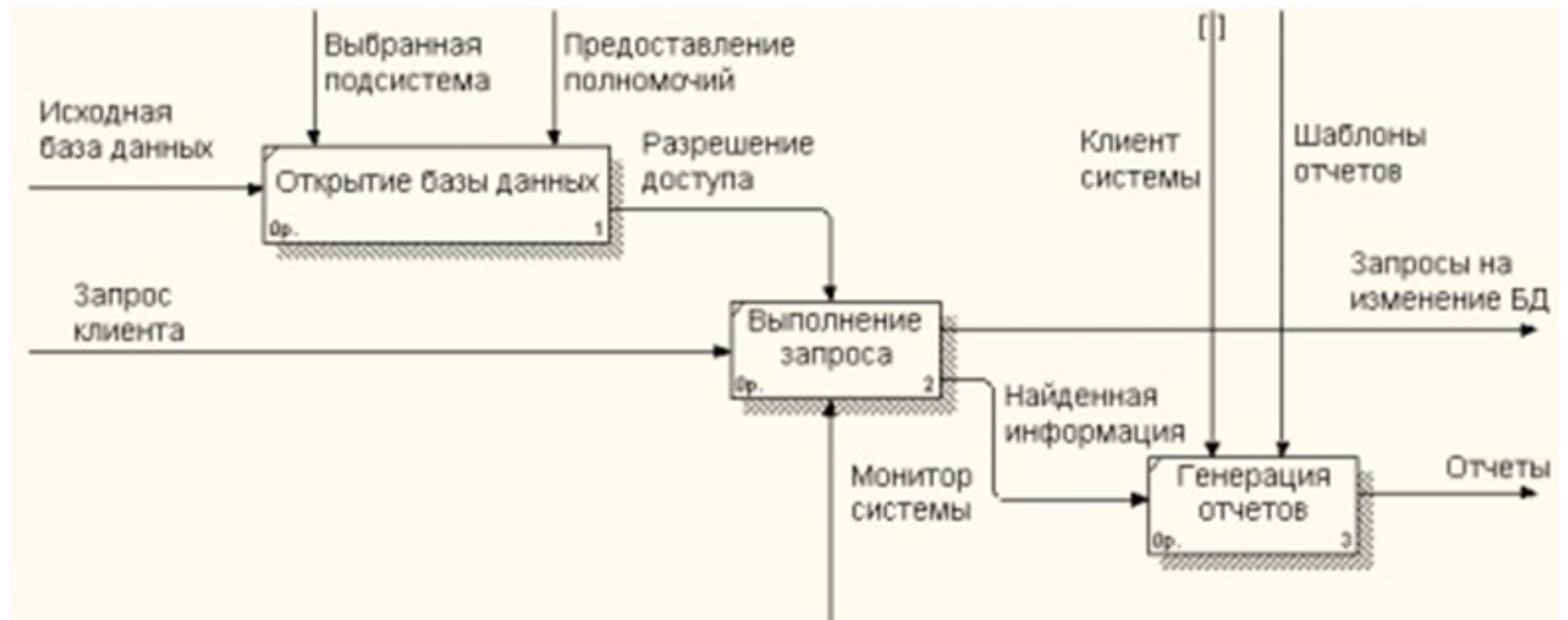
Изменение диаграммы потянет за собой корректировку всех родительских диаграмм

Корректировка диаграммы

Декомпозицию работы «Выполнение запроса» целесообразно провести при помощи диаграммы DFD, так как методология IDEF0 рассматривает систему как совокупность взаимосвязанных работ, что плохо отражает процессы обработки информации.



«Обработка запроса клиента»



Корректировка диаграммы. Декомпозиция работы

«Обслуживание клиента системы»



Корректировка диаграммы.

ІТМО

Декомпозиция контекстной диаграммы системы



Дополнение моделей процессов диаграммами DFD

DFD описывает:



- функции-обработки информации (работы);
- документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке информации;
- внешние ссылки (external reference), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
- таблицы для хранения документов (хранилища данных, data store).

Для построения диаграмм DFD обычно используется нотация Гейна -Сарсона.

Дополнение моделей процессов диаграммами DFD



Хранилище



Процесс



Внешняя сущность

Потоки данных являются механизмами, используемыми для моделирования передачи информации (или физических компонентов) из одной части системы в другую.

Потоки изображаются на диаграмме именованными стрелками, ориентация которых указывает направление движения информации. Стрелки могут подходить к любой грани прямоугольника работы и могут быть двунаправленными для описания взаимодействия типа «команда-ответ».

Назначение процесса состоит в продуцировании выходных потоков из входных в соответствии с действием, задаваемым именем процесса.

Каждый процесс должен иметь уникальный номер для ссылок на него внутри диаграммы. Этот номер может использоваться совместно с номером диаграммы для получения уникального индекса процесса во всей модели.

Дополнение моделей процессов диаграммами DFD

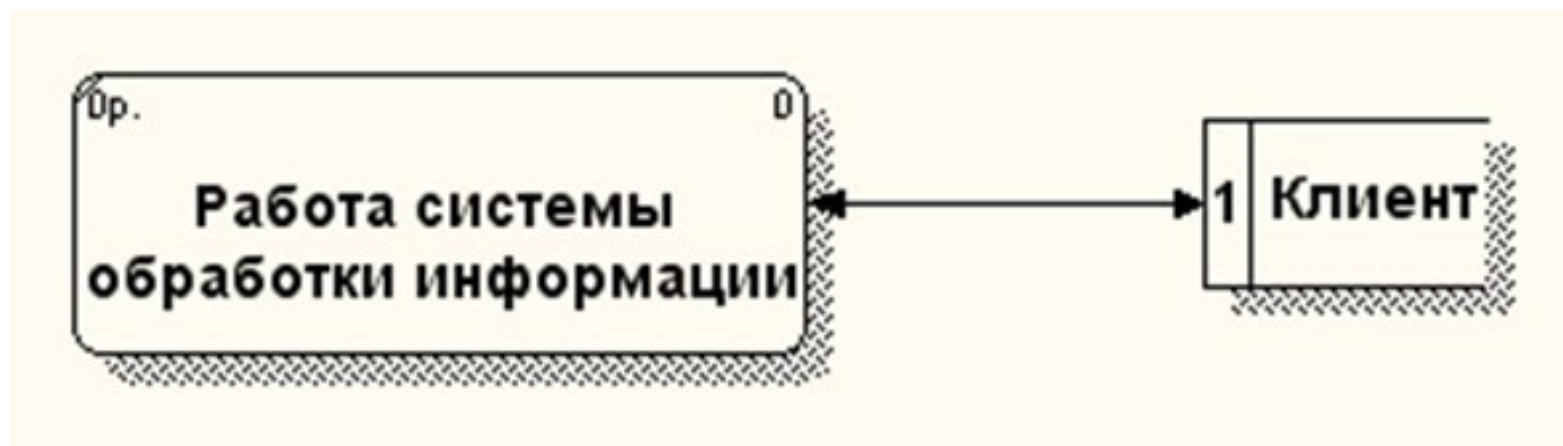
Хранилище данных позволяет на определенных участках определять данные, которые будут сохраняться в памяти между процессами.

Хранилище представляет «срезы» потоков данных во времени. Информация, которую оно содержит, может использоваться в любое время после ее определения, при этом данные могут выбираться в любом порядке. Имя хранилища должно идентифицировать его содержимое. В случае, когда поток данных входит в хранилище или выходит из него и его структура соответствует структуре хранилища, он должен иметь то же самое имя, которое не обязательно отражать на диаграмме.

Внешняя сущность представляет сущность вне контекста системы, являющуюся источником или приемником данных системы. Предполагается, что объекты, представленные такими узлами, не должны участвовать ни в какой обработке. Внешние сущности изображаются в виде прямоугольника с тенью и обычно располагаются по краям диаграммы. Одна внешняя сущность может быть использована многократно на одной или нескольких диаграммах.

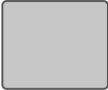
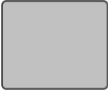








Пример

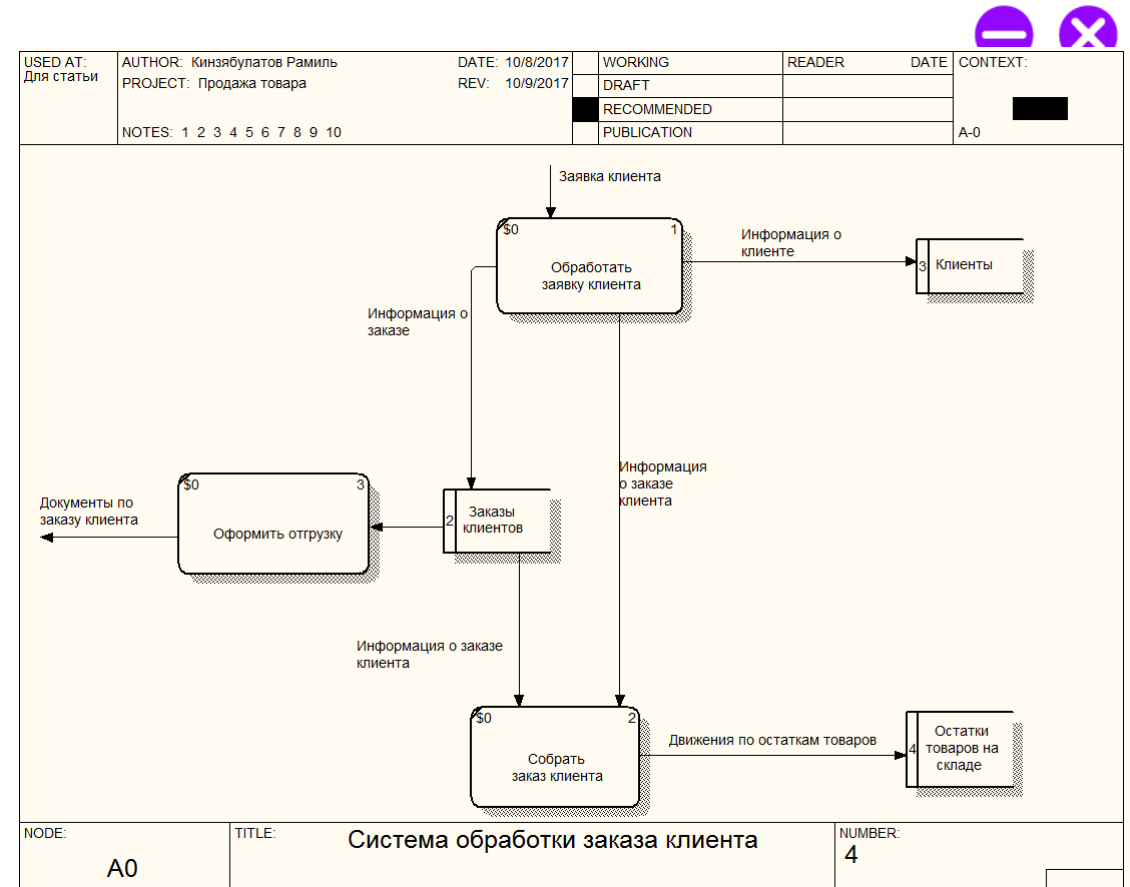
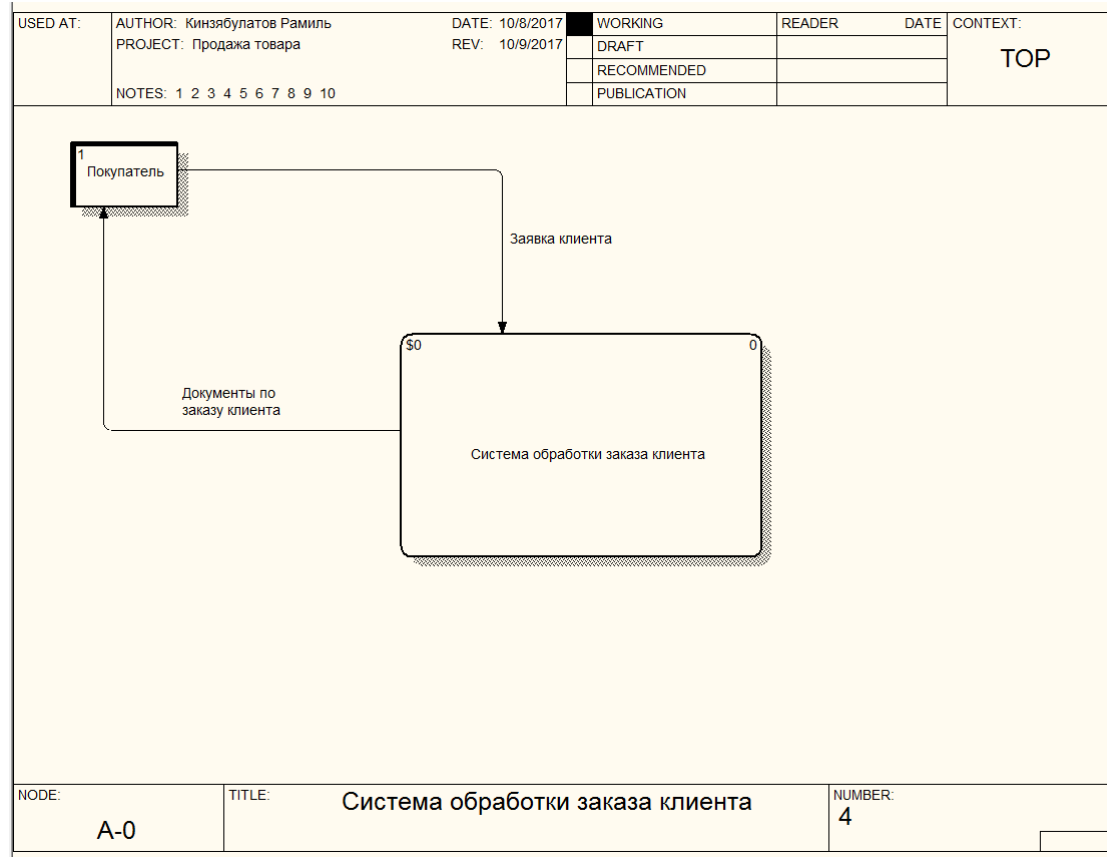


Синтаксис DFD нотации

Исторически синтаксис DFD нотации применяется в двух вариантах — Йордана (Yourdon) и Гейна-Сарсона (Gane-Sarson).

| Нотация | Юрдан и Коад | Гейн и Сарсон |
|------------------|---|---|
| Внешняя сущность |  |  |
| Процесс |  |  |
| Хранилище данных |  |  |
| Поток данных |  |  |

Примеры



Примеры

