**Лабораторная работа №5-6**

**Разработка функциональной модели в нотации IDEF0**

**Цель работы:** разработать функциональную модель с использованием нотации IDEF0 для описания функционирования информационной системы.

**Общие сведения о методологии IDEF0**

Создание современных информационных систем представляет собой сложнейшую задачу, решение которой требует применения специальных методик и инструментов. Неудивительно, что в последнее время среди системных аналитиков и разработчиков значительно вырос интерес к CASE (Computer-Aided Software Engineering)–технологиям и инструментальным CASE–средствам, позволяющим максимально систематизировать и автоматизировать все этапы разработки программного обеспечения. СASE–средства значительно облегчают задачу создания информационной системы, позволяя осуществить декомпозицию сложной системы на более простые с тем, чтобы каждая из них могла проектироваться независимо, и для понимания любого уровня проектирования достаточно было оперировать с информацией о немногих ее частях.

Стандарт IDEF0 предназначен для функционального моделирования. Его применение — это сравнительно новое направление, но уже достаточно популярное и заслужившее серьезное отношение к себе. В основе стандарта лежит понятие функции, под которой понимается управляемое действие над входными данными, осуществляющееся посредством определенного механизма, результатом его являются выходные данные.

Стандарт IDEF0 базируется на трех основных принципах:

1. Принцип функциональной декомпозиции – любая функция может быть разбита на более простые функции;
2. Принцип ограничения сложности – количество блоков от 2 до 8 – условие удобочитаемости;
3. Принцип контекста – моделирование делового процесса начинается с построения контекстной диаграммы, на которой отображается только один блок — главная функция моделирующей системы.

Специализированным средством создания IDEF0 диаграмм является Ramus Educational. Модель в Ramus Educational представляет собой совокупность SADT (Structured Analysis and Design Technique) – диаграмм, каждая из которых описывает отдельный процесс в виде разбиения его на шаги и подпроцессы. С помощью соединяющих дуг описываются объекты, данные и ресурсы, необходимые для выполнения функций. Результатом применения методологии SADT является модель, которая состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга.

Диаграммы – главные компоненты модели, все функции и интерфейсы на них представлены как блоки и дуги. Место соединения дуги с блоком определяет тип интерфейса. Диаграммы строятся при помощи блоков (см. рис.1). Каждый блок описывает какое-либо законченное действие.

Четыре стороны блока имеют различное предназначение. Слева отображаются входные данные – исходные ресурсы для описываемой блоком функции (исходная информация, материалы). Справа показываются выходные ресурсы – результирующие данные, полученные в результате выполнения описываемой блоком функции. Сверху – управление – то, что воздействует на процесс выполнения, описываемой блоком функции, и позволяет влиять на результат выполнения действия (средства управления, люди). Механизм изображается снизу — это то, посредством чего осуществляется данное действие (станки, приборы, люди и т.д.).

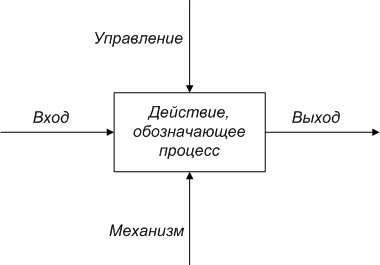


Рис.1. Построение диаграммы

Построение SADT–модели начинается с представления всей системы в виде простейшей компоненты — одного блока и дуг, изображающих интерфейсы с функциями вне системы. Поскольку единственный блок представляет всю систему как единое целое, имя, указанное в блоке, является общим. Это верно и для интерфейсных дуг — они также представляют полный набор внешних интерфейсов системы в целом. Затем блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных интерфейсными дугами. Эти блоки представляют основные подфункции исходной функции.

Данная декомпозиция выявляет полный набор подфункций, каждая из которых представлена как блок, границы которого определены интерфейсными дугами. Каждая из этих подфункций может быть декомпозирована подобным образом для более детального представления. Во всех случаях каждая подфункция может содержать только те элементы, которые входят в исходную функцию. Кроме того, модель не может опустить какие–либо элементы, нельзя ничего добавить или удалить.

Модель SADT представляет собой серию диаграмм с сопроводительной документацией, разбивающих сложный объект на составные части, которые представлены в виде блоков. Детали каждого из основных блоков показаны в виде блоков на других диаграммах. Каждая детальная диаграмма является декомпозицией блока из более общей диаграммы. На каждом шаге декомпозиции более общая диаграмма называется родительской для более детальной диаграммы.

Дуги, входящие в блок и выходящие из него на диаграмме верхнего уровня, являются точно теми же самыми, что и дуги, входящие в диаграмму нижнего уровня и выходящие из нее, потому что блок и диаграмма представляют одну и ту же часть системы. 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 (см. рис. 2).

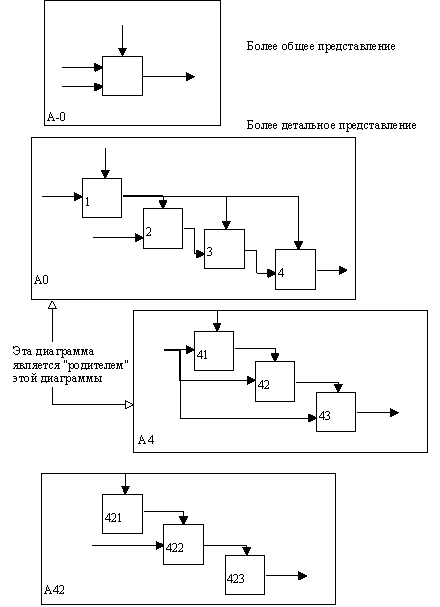
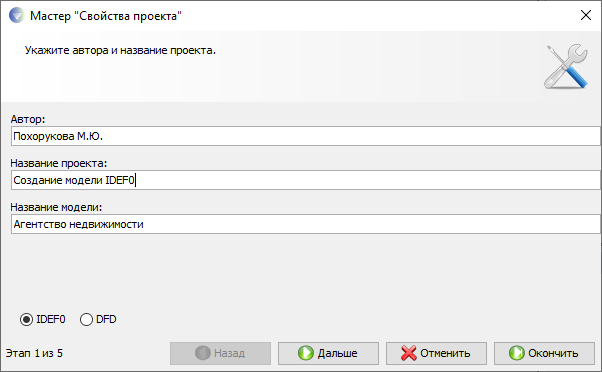


Рис. 2. Декомпозиция диаграммы

**Выполнение работы**

Рассмотрим пример построения диаграммы IDEF0 на примере информационной системы для агентства недвижимости. В системе осуществляется подбор недвижимости на основе предпочтений и пожеланий клиента для совершения сделки (купля/продажа или аренда недвижимости). В системе имеются инструменты для удобной работы с базой данных недвижимости (добавление, редактирование, удаление данных, поиск недвижимости по различным критериям, оформление договора).

Запускаем программу Ramus Educational. В открывшемся окне выбираем «создать новый файл» и нажимаем кнопку «ок». Затем указываем необходимые свойства проекта (см. рис.3).



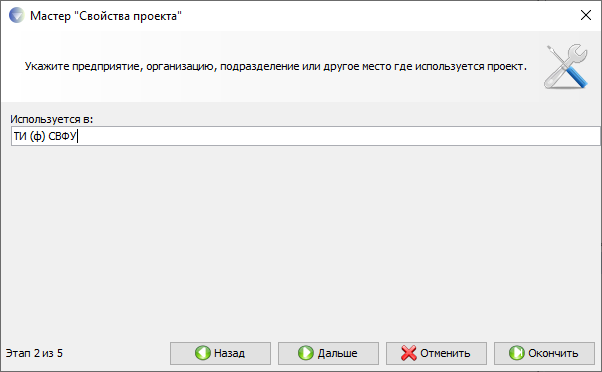


Рис. 3. Создание проекта в программе Ramus Educational

После указания этих свойств, нажимаем кнопку «Окончить». Затем выбираем «Файл»→«Сохранить как» и указываем место сохранения и имя файла. В появившемся окне проектируем модель с использованием основных инструментов. Для этого нужно выбрать какой–либо инструмент, а затем щелкнуть левой клавишей мыши в области диаграммы. Для того чтобы изменить название интерфейсных дуг и основных блоков, необходимо дважды щелкнуть по ним левой клавишей мыши. В появившемся окне также можно указать дополнительные свойства: шрифт, цвет, тип и др.

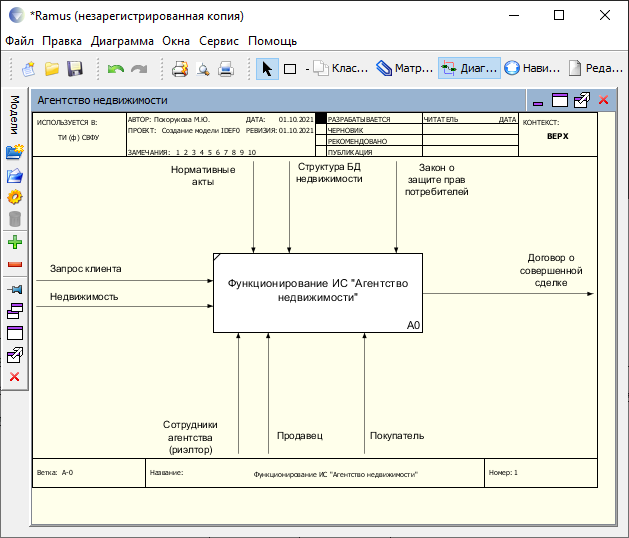
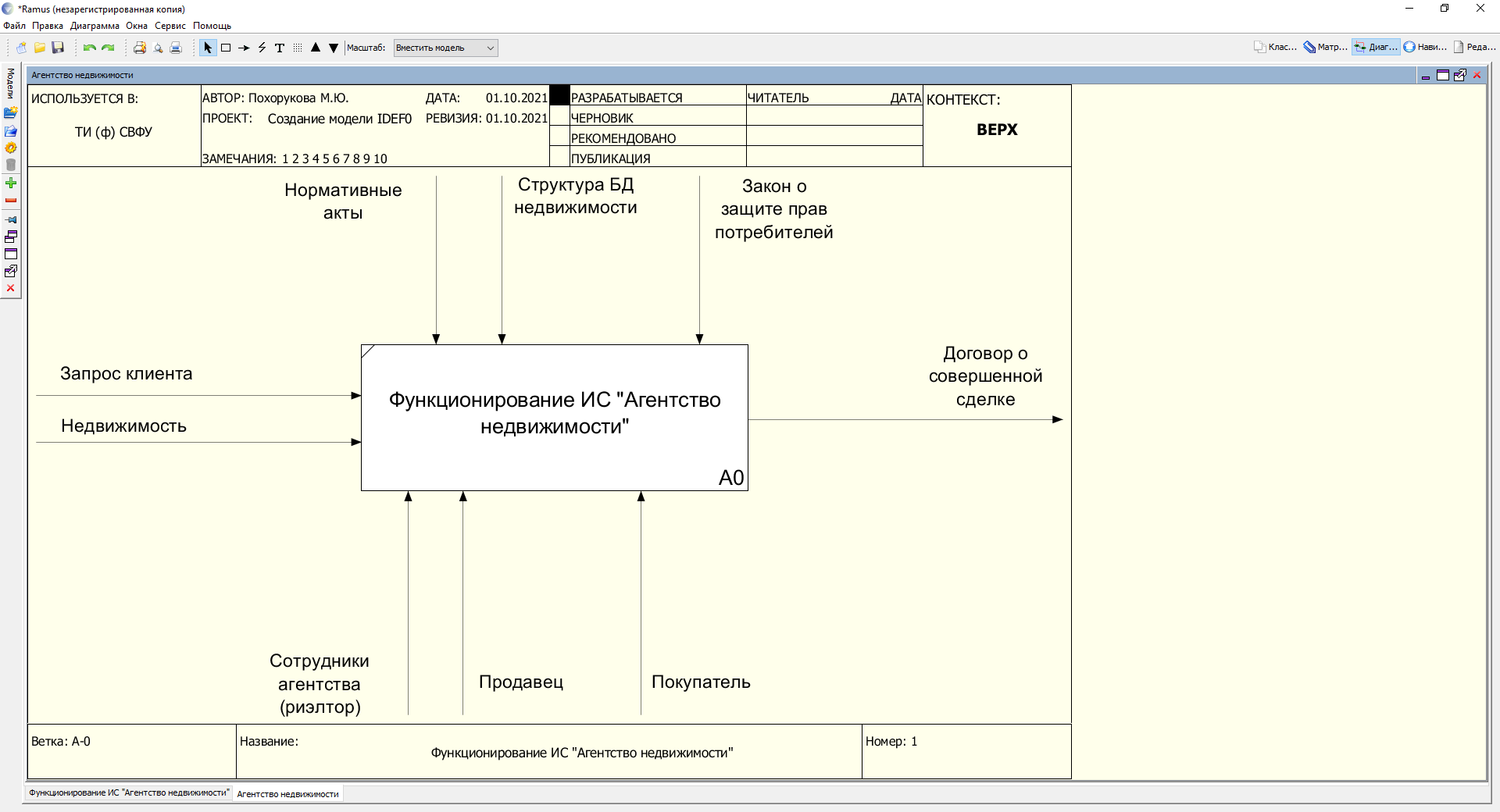
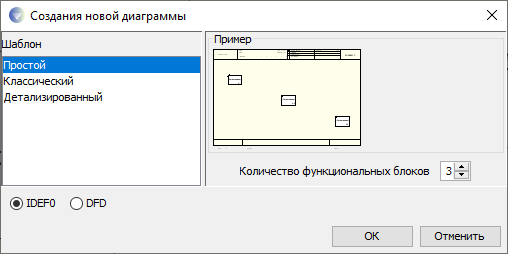


Рис. 4. Контекстная диаграмма (диаграмма 0-го уровня)

Для того чтобы декомпозировать данный процесс (разбить его на несколько более мелких подпроцессов) необходимо выделить блок и нажать на кнопку «Перейти к дочерним диаграммам»  на панели инструментов. В появившемся окне необходимо указать количество функциональных блоков и простой шаблон создания диаграмм (см. рис. 5).

   
Рис. 5. Создание дочерних диаграмм

Нажимаем кнопку «ок», в открывшемся окне заполняем все блоки и распределяем интерфейсные дуги (см. рис.6).

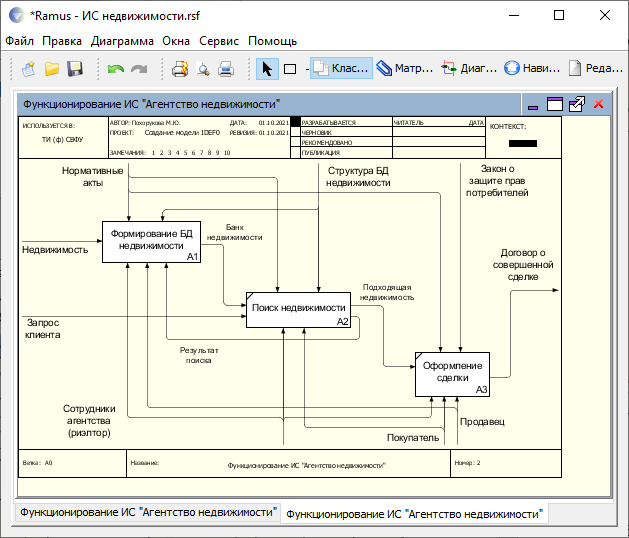


Рис. 6. Диаграмма 1-го уровня (декомпозиция контекстной диаграммы)

Так можно декомпозировать каждый из данных блоков (см. рис. 7).

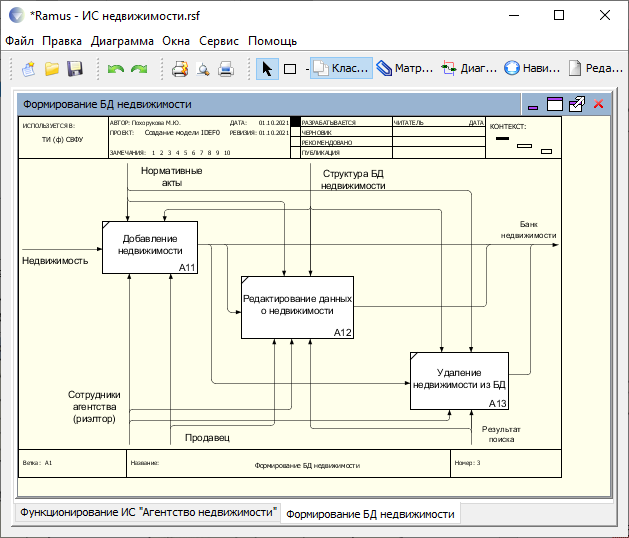


Рис. 7. Детализация (декомпозиция) блока «Формирование БД недвижимости»

**Задание:** разработайте функциональную диаграмму IDEF0 для Вашего варианта задания, ВСЕ блоки на диаграмме 1 уровня должны быть декомпозированы.

**Контрольные вопросы**

1. Что представляет собой SADT?

2. Какие основные компоненты диаграмм в нотации IDEF0?

3. Какой порядок построения функциональных диаграмм?

4. Какие существуют правила декомпозиции диаграмм? Для чего нужно декомпозировать?

5. Каково назначение дуг в диаграммах?

6. Какие существуют виды дуг?