

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Шебзухова Татьяна Александровна

Должность: Директор Пятигорского института (филиал) Северо-Кавказского

федерального университета

Дата подписания: 06.09.2023 12:54:35

Уникальный программный ключ:

d74ce93cd40e39275c3ba2f58486412a1c8ef96f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Пятигорский институт (филиал) СКФУ
Колледж Пятигорского института (филиал) СКФУ

УТВЕРЖДАЮ

Директор Пятигорского института
(филиал) СКФУ
Т.А. Шебзухова

ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ

**МДК 02.02 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ
ПРОГРАММНОГОБЕСПЕЧЕНИЯ**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

Специальности СПО

09.02.07 Информационные системы и программирование

Пятигорск 2022 г.

Методические указания для практических занятий по дисциплине МДК 02.02
Инструментальные средства разработки программного обеспечения составлены в соответствии с требованиями ФГОС СПО. Предназначены для студентов, обучающихся по специальности 09.02.07 Информационные системы и программирование.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1

Тема: История развития .Создание нового проекта MS Project.

Цель: создание проекта и заполнение базовой информации о проекте.

Порядок

1. Создан
2. Заполн
3. Изме
4. Включе ации

Пояснени

Новый проект в программе MS Project может быть создан как с нуля, так и используя один из предлагаемых стандартных шаблонов. Шаблон представляет собой особенный тип файла проекта, содержащий набор информации, призванной упростить работу над проектом. В состав шаблона обычно входит список заранее организованных и размещенных определенным образом задач, а также информация о ресурсах, пользовательские представления, календари, отчеты, макросы и т.д. Любая информация, предлагаемая шаблоном, может быть изменена в соответствии с требованиями конкретного проекта. В качестве шаблона также может быть использован созданный ранее проект. При создании проекта из шаблона необходимо выбрать на панели *Консультанта* ссылку *Общие шаблоны*. Далее на вкладке *Шаблоны проектов* выбирается необходимый шаблон.

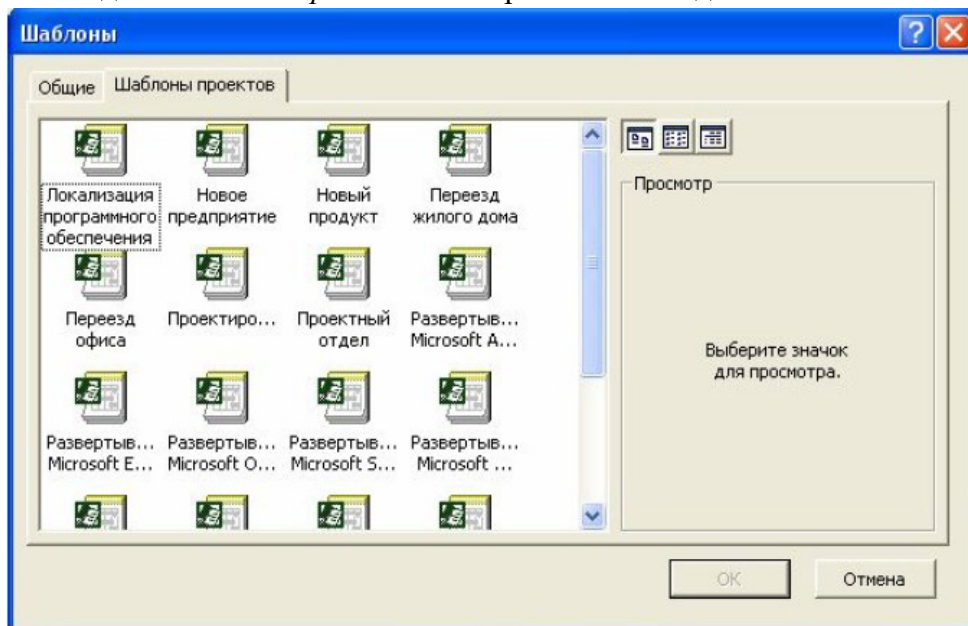


Рис.1. Выбор шаблона проекта

Рабочее пространство программы называется видом или представлением. По умолчанию после создания проекта активен вид *Диаграмма Ганта* (рис.2). Данная диаграмма служит для отображения последовательности задач проекта как в текстовом так и в графическом виде.

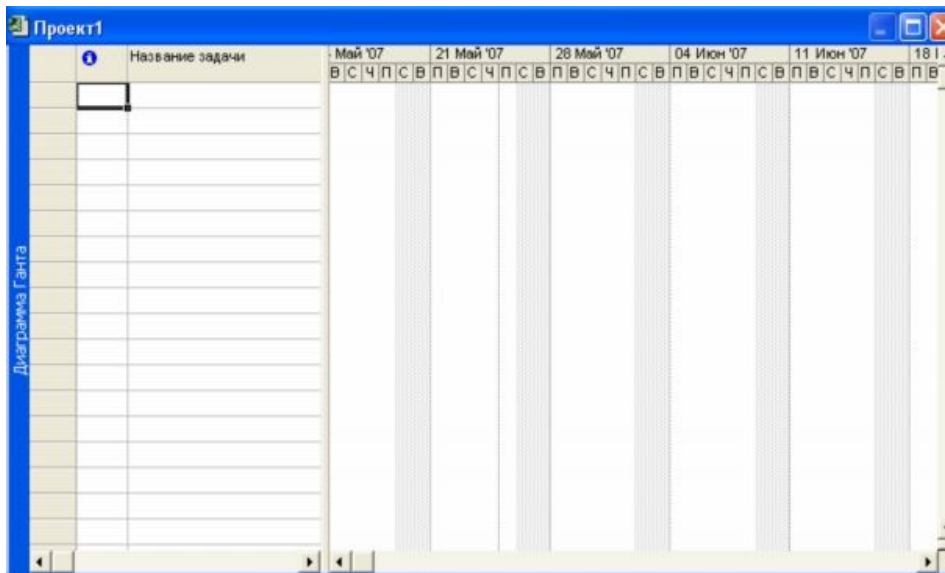


Рис.2. Окно диаграммы Ганта

После создания проекта необходимо настроить его основные параметры. Для этого удобно использовать мастер *Новый проект*. Для этого нажимаем кнопку *Задачи* на панели *Консультанта* и выбираем ссылку *Определение проекта*. Отвечив на вопросы о дате начала проекта и совместной работе над проектом и сохранив результат, выбираем ссылку *Определение рабочего времени проекта* для запуска мастера *Рабочее время проекта*. Таким образом мы можем настроить календарь проекта. Следующим решением, которое необходимо принять на стадии создания, является выбор исходной даты проекта. План проекта может быть составлен от даты начала или завершения проекта. Для настройки планирования от начальной даты выберите в меню *Проект* пункт сведения о проекте. В появившемся окне (рис.3) выбираем планирование *От даты начала проекта* и ставим *Дату начала*. Да окончания будет рассчитана далее автоматически. В случае планирования от конечной даты выбираем *От даты окончания проекта* и ставим *Дату окончания*. В этом случае автоматически будет рассчитываться дата начала.

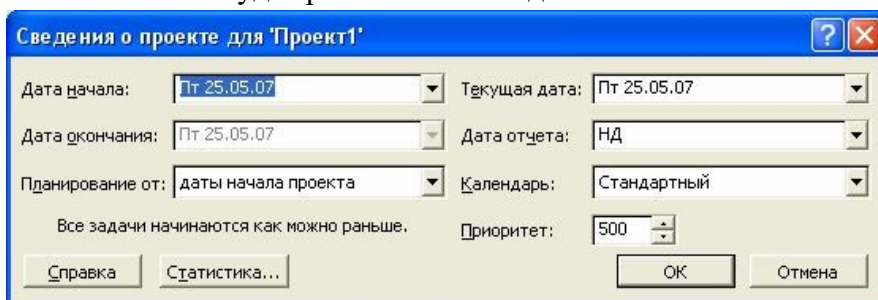


Рис.3. Настройка сведений о проекте

Также в этом окне мы можем выбрать календарь для проекта. В состав пакета MS Project входит три базовых календаря – стандартный, ночная смена и 24 часа. В *стандартном календаре* рабочий день начинается с 8:00 и заканчивается в 17:00

с обеденным перерывом с 12:00 до 13:00. Рабочая неделя начинается с понедельника и заканчивается в пятницу. Это календарь, применяемый по умолчанию. В *календаре ночной смены* рабочий день начинается с 23:00 и заканчивается в 8:00 с часовым перерывом с 03:00 до 04:00. В *календаре «24 часа»* рабочее время продолжается круглые сутки без выходных и обеденных перерывов. Базовые календари можно редактировать для этого в меню *Сервис* необходимо выбрать пункт *Изменение рабочего времени*. В появившемся окне (рис.4.) выбираем базовое расписание, которое мы хотим отредактировать. Для изменения рабочего времени одного дня необходимо выбрать этот день в календаре. Далее, если необходимо сделать этот день выходным, мы выбираем параметр *нерабочее время*, если же мы хотим только изменить временные рамки рабочего дня, то выбираем параметр *нестандартное*

рабочее время и в полях ниже вводим время начала и завершения рабочего дня.

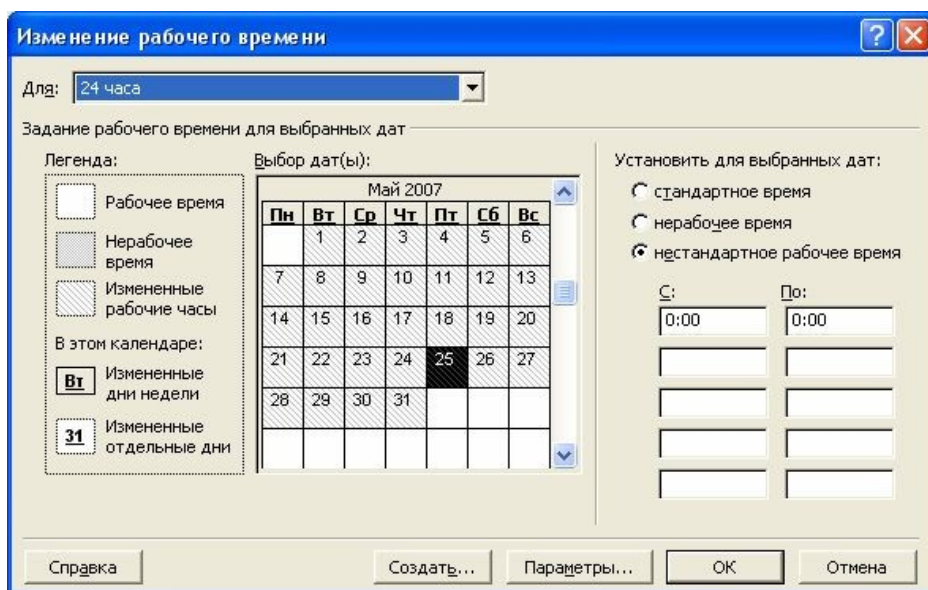


Рис.4. Изменение рабочего времени.

Можно также создать новое базовое расписание. Для этого в окне *Изменение рабочего времени* нажимаем кнопку *Создать*. В появившемся окне (рис.5) выбираем создание нового календаря на основе стандартного или создание копии любого другого календаря. Значения рабочего времени для вновь созданного календаря могут также быть отредактированы через окно *Изменение рабочего времени*.

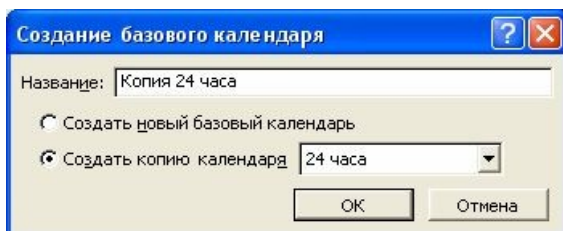
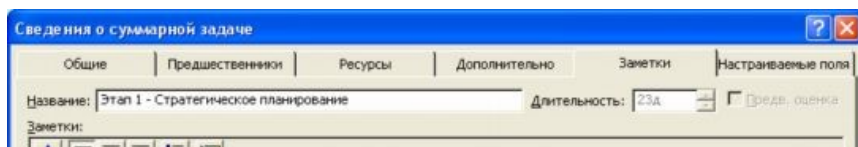


Рис.5. Создание базового календаря.

Создаваемый проект может быть использован в качестве хранилища проектной документации, например обзора проекта, результатов проведенных анализов или спецификации создаваемого продукта. Для присоединения такой документации целесообразно использовать т.н. суммарную задачу проекта, содержащую итоговую информацию о датах и стоимости проекта. Для отображения суммарной задачи на диаграмме Ганта необходимо в меню *Сервис* выбрать пункт *Параметры* и перейти на вкладку *Вид*. На данной вкладке необходимо выбрать параметр *Показать суммарную задачу проекта* под заголовком *Параметры структуры для проекта*. Суммарная задача появится в нулевом ряду диаграммы Ганта. Проектная документация может как включаться в файл проекта, так и быть доступной через гиперссылки. Для включения документов в файл проекта необходимо выбрать суммарную задачу проекта и нажать кнопку *Сведения о задаче*, расположенную на стандартной панели задач. В открывшемся окне (рис.6) выбираем вкладку *Заметки*. На вкладке нажимаем кнопку *Вставить объект*. В открывшемся окне необходимо выбрать опцию *Создать из файла*. После этого указываем путь к файлу документа, который предполагается

включить

в проект.



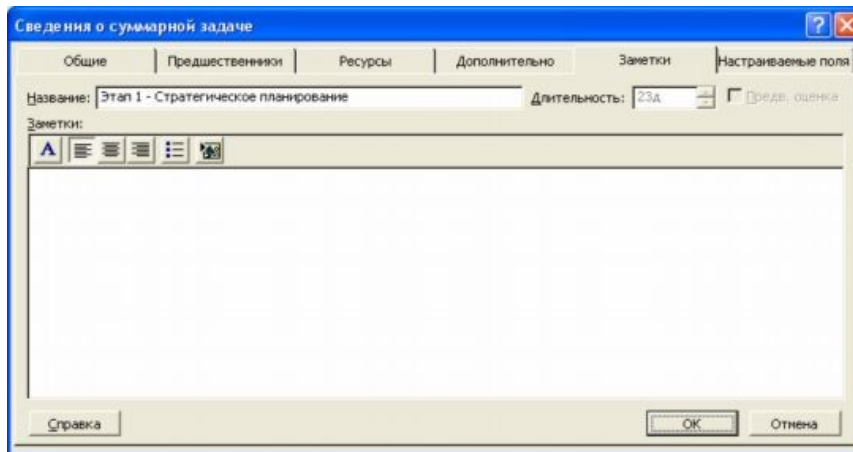


Рис.6. Сведения о задаче.

После закрытия окна сведений о суммарной задаче в диаграмме Ганта появится индикатор примечаний. Для создания гиперссылки к документу необходимо нажать кнопку Гиперссылка на панели задач. В поле Текст открывшегося диалогового окна *Добавление гиперссылки* (рис.7) введите название связываемого документа, затем выберите документ в списке. В поле индикаторов диаграммы Ганта появится индикатор гиперссылок.



Рис.7. Добавление гиперссылки.

Контрольные вопросы:

1. Зачем необходимы шаблоны проектов?
2. В чем разница между планированием проекта от даты начала или даты его окончания?
3. Какие существуют базовые календари в программе MS Project?
4. Как внести изменения в базовый календарь?
5. Как включить в проект проектную документацию?

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2

Тема: Базовые принципы построения Case средств. Расписание проекта

Цель: создание развернутого расписания работ.

Порядок выполнения

1. Внести задачи проекта в табличную часть диаграммы Ганта.
2. Упорядочить задачи проекта в соответствии с логикой проекта.
3. Ввести кода структурной декомпозиции работ.

Пояснения к выполнению

Для создания расписания работ в программе MS Project необходимо внести задачи в диаграмму Ганта в соответствии с их иерархией. Для перехода в режим диаграммы Ганта необходимо выбрать из меню *Вид* пункт *Диаграмма Ганта*.

Существуют несколько методик внесения задач в диаграмму Ганта:

- *Произвольный ввод* – задачи вносятся без соблюдения последовательности или группировки задач. Необходимые изменения вносятся позже;
- *Последовательный ввод* – задачи вводятся последовательно, от начала до завершения проекта (или наоборот);
- *Обозначение фаз* – вносятся только главные задачи. Далее следует их декомпозиция;
- *Обозначение вех* - внесение ключевых задач. Далее в расписание вносятся задачи, необходимые для выполнения ранее внесенных задач. Для добавления задачи в столбец *Название задачи* вводится название задачи. В дальнейшем название и другие свойства задачи можно редактировать двойным нажатием на любом столбце табличной части диаграммы Ганта. Данные о задачах также могут быть импортированы из программного продукта MS Excel. Для этого необходимо, чтобы поля таблиц MS Excel полностью соответствовали полям диаграммы Ганта. Поэтому сначала создаем на основе *Шаблона импорта списка задач Microsoft Project* новую таблицу MS Excel (рис.8).

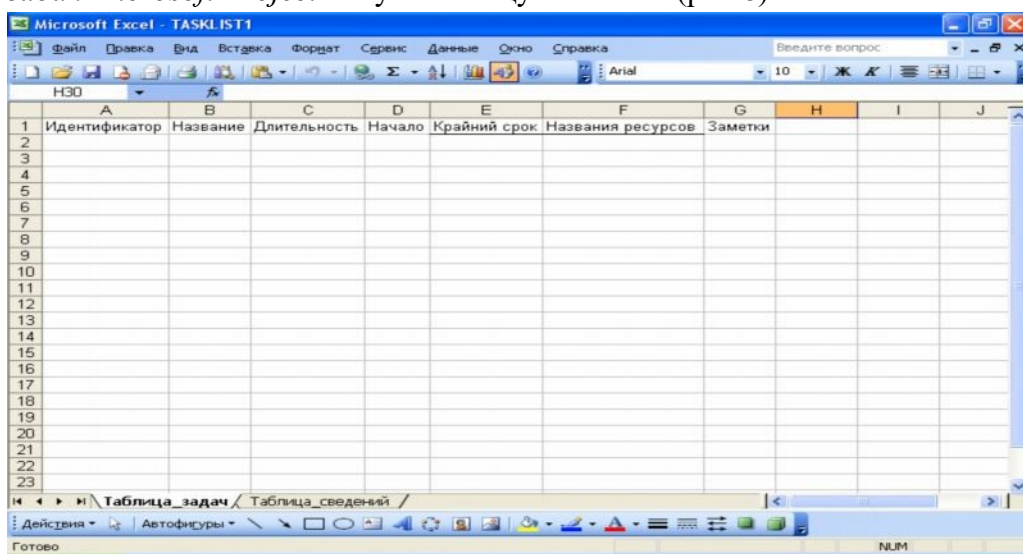


Рис.8. Шаблон импорта списка задач Microsoft Project.

После заполнения и сохранения файла таблицы необходимо открыть этот файл в MS Project. Для этого, нажав в меню *Файл* пункт *Открыть*, в диалоговом окне открытия файла указываем тип файла *Книги Microsoft Excel* и выбираем файл сохраненной таблицы из списка. Некоторые задачи повторяются регулярно, например еженедельно. Для задач такого типа нет необходимости вводить несколько раз одну и ту же информацию, достаточно указать, что задача является повторяющейся. В таком случае необходимо в диаграмме Ганта выбрать задачу, после которой вы хотите вставить повторяющуюся задачу. Затем в меню *Вставка* выбрать пункт *Повторяющаяся задача*. В открывшемся окне (рис.9) необходимо заполнить название задачи, указать частоту и диапазон повторений. Также в случае необходимости указываем дату окончания задачи.

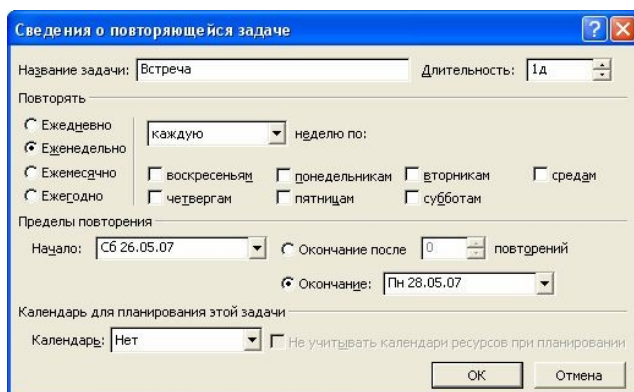


Рис.9. Сведения о повторяющейся задаче.

В диаграмме Ганта повторяющаяся задача отмечена специальным индикатором .

Повторяющаяся задача показана в виде суммарной задачи, где все ее повторения являются отдельными подзадачами. Для расположения задач в логической последовательности может быть использовано перемещение, добавление, копирование и удаление задач. Для *перемещения задачи* строку задачи в диаграмме Ганта необходимо выделить целиком, для этого нажимаем на заголовке строки серого цвета. После изменения курсора на «+» перетаскиваем строку в желаемое место. Для *добавления задачи* выбираем строку в диаграмме Ганта, выше которой будет располагаться новая задача и нажимаем в меню *Вставка* пункт *Новая задача* либо нажимаем кнопку *Insert*. Для *копирования задачи* необходимо выбрать задачу и нажать кнопку *Копировать ячейку* на стандартной панели инструментов. Далее, перейдя в свободную строку, нажимаем кнопку *Вставить*. Можно скопировать одновременно несколько задач, для этого нажав *Ctrl* (для несмежных задач) или *Shift* (для задач, расположенных рядом), выделяем необходимые задачи. Для *удаления задачи* необходимо выбрать соответствующую строку таблицы и нажать *Delete*. После расположения задач в логической последовательности необходимо создать структуру, представляющую иерархию выполняемых задач. Задача, расположенная на самом верхнем уровне структуры расписания, называется суммарной задачей. Задачи более низкого уровня называются подзадачами. Каждая такая подзадача, в свою очередь, может быть также разделена на подзадачи. MS Project поддерживает до девяти уровней вложенности задач. Для структурирования задач можно использовать следующие средства MS Project:

- *Перемещение задачи на один уровень ниже.* Для этого необходимо выделить перемещаемую задачу в диаграмме Ганта и нажать кнопку *На уровень ниже* на панели инструментов *Форматирование*. Выбранная задача становится подзадачей, а вышестоящая становится для нее суммарной. Перемещать можно и несколько задач, предварительно выделив их с помощью *Ctrl* или *Shift*.

- *Перемещение задачи на один уровень выше.* Для этого необходимо выбрать перемещаемую задачу и нажать кнопку *На уровень выше* на панели инструментов *Форматирование*.

- *Отображение всех задач, вплоть до указанного уровня вложенности* Для этого необходимо нажать кнопку *Показать* на панели инструментов *Форматирование*. Из списка выбрать необходимый уровень. Будут отображаться только задачи данного уровня или более высоких.

- *Скрыть или показать все подзадачи для данной задачи.* Для этого необходимо нажать знак «плюс» или «минус» слева от заголовка задачи.

После структуризации можно настроить *кода структурной декомпозиции работ* (СДР). Каждый уровень и элемент структурной декомпозиции работ описывается с помощью уникального кода. Как правило, каждая цифра, присутствующая в таких кодах, указывает на уровень в структурной иерархии. По умолчанию, MS Project создает кода СДР на основе структурных номеров. Для отображения структурных номеров в диаграмме Ганта необходимо нажав правой кнопкой на заголовке табличной части диаграммы выбрать пункт меню *Вставить столбец*. В открывшемся окне (рис.10) выбираем имя поля *Номер в структуре*.

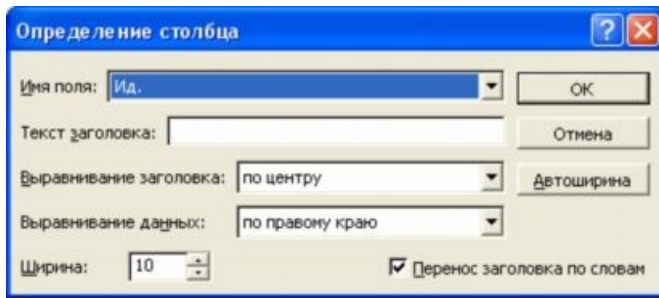


Рис.10. Определение столбца.

Для настройки собственной схемы СДР необходимо в меню *Проект* выбрать пункт *СДР/Определить код*. В открывшемся окне (рис.11) выбираем префикс кода проекта, далее в поле *Последовательность* выбираем формат кода для уровней иерархии, начиная с первого. После закрытия окна в табличную часть диаграммы Ганта вставляем столбец с заголовком *СДР*.

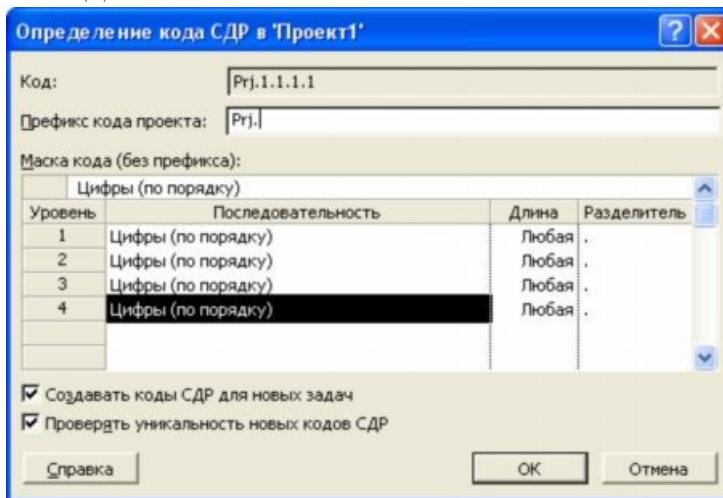


Рис.11. Определение кода СДР в проекте.

Контрольные вопросы:

1. Что такое расписание проекта?
2. Как отображаются задачи в диаграмме Ганта?
3. Как провести импорт задач проекта из MS Excel?
4. Как добавить в проект повторяющуюся задачу?
5. Какие существуют в MS Project средства для управления иерархией проекта?
6. Для чего предназначены коды структурной декомпозиции работ?

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3

Тема: Основные функциональные возможности. Временные связи

Цель: расстановка временных связей между задачами.

Порядок выполнения

1. Установить длительность всех дочерних задач.
2. Указать зависимости задач.
3. Настроить ограничения задач.
4. Настроить напоминания о крайних сроках.
5. Добавить вехи.
6. Настроить календари задач.

Пояснения к выполнению

После заполнения списка задач и упорядочивания его, необходимо внести информацию о количестве рабочего времени, необходимого для завершения каждой задачи. Для этого необходимо внести значение продолжительности задачи в поле *Длительность* табличной части диаграммы Ганта, например « 1 день». Значение длительности также может быть внесено как приблизительное. MS Project рассчитывает такую длительность как и подтвержденную, однако данная возможность позволяет указать предварительный характер

длительности, который необходимо уточнить по ходу разработки проекта. Для внесения приблизительной длительности необходимо после временной единицы добавить знак вопроса, например «1 нед?» (рис.12). Удалив знак вопроса, приблизительную длительность задачи можно изменить на подтвержденную.

		Название задачи	Длительность	Начало	Окончание	Предш
1		☐ Шаблон локализации продукта	146 дней?	Чт 01.01.04	Чт 22.07.04	
2		☐ Область охвата	25 дней?	Чт 01.01.04	Ср 04.02.04	
3		Изучение задачи	2 дней	Чт 01.01.04	Пт 02.01.04	
4		Оценивание требуемых усилий (на приме	1 нед	Пн 05.01.04	Пт 09.01.04	3
5		Подготовка спецификации локализации (о	2 дней	Пн 12.01.04	Вт 13.01.04	4
6		Предварительная оценка затрат	3 дней	Ср 14.01.04	Пт 16.01.04	5
7		Определение рисков	2 дней?	Пн 19.01.04	Вт 20.01.04	6
8		Закрепление финансирования проекта	1 день	Ср 21.01.04	Ср 21.01.04	7
9		Предварительное определение ресурсов	1 нед	Чт 22.01.04	Ср 28.01.04	8
10		Определение матрицы и способов обмена	3 дней	Чт 22.01.04	Пн 26.01.04	8
11		Закрепление ключевых ресурсов	1 нед	Чт 29.01.04	Ср 04.02.04	9
12		Область охвата завершена	0 дней	Ср 04.02.04	Ср 04.02.04	11;10

Рис.12. Длительность задач

Для задания длительности в проекте могут использоваться различные временные единицы, включая:

- минуты (мин);
- часы (ч);
- дни (д);
- недели (нед);
- месяцы (мес).

При вводе длительности задач могут использоваться как полные названия временных единиц, так и их аббревиатуры. Если единицу измерения времени не указывать, MS Project автоматически будет использовать дни.

Для получения достоверной информации о длительности задач можно использовать четыре возможных источника:

- Знания сотрудников, работающих над проектом;
- Экспертные оценки;
- Информация о предыдущих проектах;
- Промышленные стандарты.

При существовании расхождений в оценке длительности задач или необходимости моделирования альтернативных сценариев выполнения проекта используется методика PERT (Program, Evaluation and Review Technique). При анализе методом PERT для расчета длительности задачи используются средневзвешенные значения оптимистической, пессимистической и ожидаемой длительности. Анализ PERT является эффективным методом предупреждения рисков. Он позволяет рассчитывать расписание проекта с учетом возможного или имеющегося времени, ресурсов или стоимости.

Для выполнения анализа методом PERT необходимо выбрать пункт *Анализ по методу PERT* в меню *Вид/Панели инструментов*. На появившейся панели инструментов *Анализ по методу PERT* нажмите кнопку *Лист ввода PERT* (). Введите для каждой задачи значения оптимистической, пессимистической и ожидаемой длительности в соответствующие поля таблицы (рис.13). Если длительность известна точно, то во всех полях таблицы указывается одинаковая длительность.

	Название задачи	Длительность	Оптимистическая длительность	Ожидаемая длительность	Пессимистическая длительность
1	☐ Шаблон локализации	143,33 дней?	134 дней	143 дней	154 дней
2	☐ Область охвата	22,33 дней?	13 дней	22 дней	33 дней
3	Изучение зад	2 дней	1 день	2 дней	3 дней
4	Оценивание т	1,67 нед	6 дней	8 дней	12 дней
5	Подготовка сг	2 дней	1 день	2 дней	3 дней
6	Предваритель	2 дней	1 день	2 дней	3 дней
7	Определение	2 дней?	1 день	2 дней	3 дней
8	Закрепление	2 дней	1 день	2 дней	3 дней
9	Предваритель	0,4 нед	1 день	2 дней	3 дней
10	Определение	2 дней	1 день	2 дней	3 дней
11	Закрепление	0,4 нед	1 день	2 дней	3 дней

	Название задачи	Длительность	Оптимистическая длительность	Ожидаемая длительность	Пессимистическая длительность
1	Шаблон локализаци	143,33 дней?	134 дней	143 дней	154 дней
2	Область охвата	22,33 дней?	13 дней	22 дней	33 дней
3	Изучение зад	2 дней	1 день	2 дней	3 дней
4	Оценивание т	1,67 нед	6 дней	8 дней	12 дней
5	Подготовка сг	2 дней	1 день	2 дней	3 дней
6	Предваритель	2 дней	1 день	2 дней	3 дней
7	Определение	2 дней?	1 день	2 дней	3 дней
8	Закрепление	2 дней	1 день	2 дней	3 дней
9	Предваритель	0,4 нед	1 день	2 дней	3 дней
10	Определение	2 дней	1 день	2 дней	3 дней
11	Закрепление	0,4 нед	1 день	2 дней	3 дней
12	Область охва	0 дней	0 дней	0 дней	0 дней

Рис 13. Лист ввода PERT

При нажатии на кнопку *Вычисления по методу PERT* на этой же панели инструментов программа рассчитает приблизительную длительность задачи, которая будет внесена в поле таблицы *Длительность*. Следующим шагом в построении базового расписания является установка взаимосвязей между задачами. Связь между задачами возникает в том случае, если начало и/или конец одной задачи как-либо зависит от другой задачи. В MS Project можно создать связь между предыдущей задачей, называемой *предшественником* и следующей задачей - *последователем*, и тем самым создать зависимость между ними. Для этого необходимо перейти в режим диаграммы Ганта. Выбираете две задачи, которые необходимо связать между собой. Если задачи расположены в списке задач не рядом, выделяем задачу-предшественник, а затем, удерживая клавишу Ctrl, выбираем задачу-последователь. Далее нажимаем кнопку *Связать задачи* на панели задач (). После этого связь будет показана на диаграмме Ганта стрелкой и в столбце *Предшественники* появится номер задачи-предшественника. Также связь может быть установлена указанием задачи-предшественника в столбце *Предшественники* напрямую. По умолчанию всегда строится связь типа «конец-начало», когда вторая задача начинается сразу после конца первой. Однако MS Project поддерживает и другие типы связей (рис.14). Для их создания на диаграмме Ганта выберите задачу, являющуюся последователем в устанавливаемой связи. Двойным нажатием мыши открываем окно *Сведения о задаче*. На вкладке *Предшественники* выбираем любую свободную ячейку в поле *Имя задачи*. В строке появится раскрывающийся список, содержащий задачи проекта. Выберите задачу, являющуюся предшественником в устанавливаемой зависимости. Нажав на поле *Тип* выберите из списка тип связи. Для частичного совмещения задач необходимо в поле *Запаздывание* ввести устанавливаемое время опережения. Это значение вводится либо в процентах, либо во временных единицах и для случая совмещения задач должно быть отрицательным. В случае, если при выполнении задачи-последователя возникает запаздывание, в поле *Запаздывание* указывается



положительной значение.

Рис.14. Типы связей.

Для удаления связи необходимо на вкладке *Предшественники* в окне *Сведения о задаче* для удаляемой связи указать тип связи (*Нет*). После закрытия окна связь исчезнет. Любая задача в MS Project имеет ограничения по времени выполнения. По умолчанию это *Как можно раньше*. MS Project поддерживает следующие типы ограничений:

- Гибкие – *Как можно раньше* и *Как можно позже*. Наличие таких ограничений повышает гибкость времени выполнения задачи;
- Изменяемо-гибкие – *Начало не ранее*, *Начало не позднее*, *Окончание не ранее* и *Окончание не позднее*. Такие ограничения позволяют использовать при работе с ними временной интервал и допускают наличие вариантов времени выполнения;
- Негибкие – *Фиксированное начало* и *Фиксированное окончание*. Все остальные ограничения будут второстепенными по отношению к данным. Вне зависимости от взаимосвязей задач они должны выполняться с учетом фиксации дат начала/окончания. Для изменения ограничений необходимо на вкладке *Дополнительно* окна *Сведения о задаче* (рис.15) поменять *Тип ограничения* и *Дату ограничения*. Также при установке даты начала или конца задачи в табличной части диаграммы Ганта тип ограничения также автоматически изменится на *Фиксированное* *начало/окончание*.

Рис.15. Сведения о задаче.

В тех случаях, когда задача должна быть завершена к определенному сроку, но расписание менять нежелательно, используются напоминания о крайних сроках. Такое напоминание не влияет на расписание проекта, при выполнении задачи после крайнего срока в столбце индикаторов появляется предупреждение. Для установки крайнего срока необходимо выбрать дату в поле *Крайний срок* вкладки *Дополнительно* окна *Сведения о задаче*. После этого в графической части диаграммы Ганта появится указатель крайнего срока (рис.16).



Рис.16. Указатель крайнего срока.

Также крайние сроки можно сразу ввести в столбец *Крайний срок* табличной части диаграммы Ганта. Для указания начала или завершения основных фаз проекта, либо для указания завершения назначений проекта могут использоваться *вехи*, специально выделяемые задачи, отображающие достижение промежуточных результатов проекта. Вехи также не влияют на расписание проекта, однако, как правило, они взаимосвязаны с другими задачами проекта. Простейшим способом создания вехи является ввод задачи нулевой длительности. Указатель вехи появляется на графической части диаграммы Ганта (рис.17).



Рис.17. Указатель вехи

Для ввода вех ненулевой длительности, например последней задачи, выполняемой на каждой фазе проекта, необходимо установить флажок *Пометить задачу как вежу* на вкладке *Дополнительно* окна *Сведения о задаче*, предварительно выделив желаемую задачу. В случае если необходимо планировать задачу с учетом рабочего времени, отличного от

принятого в календаре проекта или календаре ресурса, задаче назначается свой календарь. В качестве такого календаря можно выбрать один из уже имеющихся базовых календарей или создать новый. Для создания нового календаря необходимо в меню *Сервис* выбрать пункт *Изменить рабочее время*. В открывшемся окне нажимаем кнопку *Создать...* Задав имя нового календаря и выбрав исходный календарь, вносим желаемые изменения в рабочее время. Закрыв окно, мы можем назначить созданный календарь задаче. Для этого в окне *Сведения о задаче* на вкладке *Дополнительно* заполняем поле *Календарь*. После выбора календаря в поле индикаторов появляется индикатор календаря.

Контрольные вопросы:

1. Как установить длительность родительской задачи?
2. Какие существуют виды временных зависимостей задач?
3. Как настроить временные ограничения задач?
4. В чем заключаются основные функции напоминаний о крайних сроках и вех проекта?
5. Как взаимосвязаны календари проекта, ресурсов и задач?

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4

Тема: Классификация Case средств. Ресурсы

Цель: добавление в проект ресурсов.

Порядок выполнения

1. Добавить в проект трудовые ресурсы.
2. Добавить в проект материальные ресурсы.
3. Настроить календари рабочего времени ресурса.
4. Назначить ресурсы задачам.
5. Профилирование загрузки ресурсов.

Пояснения к выполнению

Выполнение любых задач обеспечивается тремя видами ресурсов – людьми, машинами и оборудованием. В программе MS Project все ресурсы делятся на две категории- люди и оборудование относятся к трудовым ресурсам. За единицу измерения усилий, затраченных такими ресурсами при выполнении задачи, принято время. Материалы относятся к материальным ресурсам. Для них измеряемой величиной по отношению к выполняемым задачам является потребление ресурса, а единицей измерения- количество ресурса. Для добавления ресурсов в проект необходимо выбрать представление листа ресурсов, для этого выбираем пункт *Лист ресурсов* в меню *Вид* (Рис.18). В открывшейся таблице вводим название ресурса и выбираем его тип.

	Название ресурса	Тип	Единицы измерения материалов	Краткое название	Группа	Макс. единиц	Стандартная ставка	Ставка сверхурочных	Затраты на использование	Начисление	Базовый календарь
1	Иванов И.И.	Трудовой		ИИИ	Менеджер	100%	200,00р./ч	350,00р./ч	0,00р.	Пропорциональное	Стандартный
2	Ст. мастер	Трудовой		С		100%	120,00р./ч	180,00р./ч	0,00р.	Пропорциональное	Стандартный
3	Стройматериалы	Материальный		СМ			150 000,00р.		0,00р.	Пропорциональное	

Рис.18. Лист ресурсов

Для трудового ресурса необходимо указать максимальное количество доступных единиц. Ресурсам, работающим на полную ставку, присваивается 100%. Далее указываем стандартную ставку оплаты труда и сверхурочную (если есть). По умолчанию используется почасовая оплата, но можно указывать и месячную ставку. Также можно указать *затраты на использование* – дополнительные одноразовые затраты, применяемые при каждом назначении ресурса. *Начисление* затрат на ресурсы может осуществляться в конце или начале задачи или пропорционально ее выполнению. По умолчанию в качестве базового календаря применяется стандартный, однако в случае особых условий труда можно выбрать любой другой календарь или создать свой. В случае если максимум доступных единиц трудового ресурса будет изменяться по ходу выполнения проекта, необходимо настроить доступность ресурса. Для этого выделяем ресурс в *Листе Ресурсов* и нажимаем кнопку *Сведения о ресурсе* на панели инструментов *Стандартная*. В открывшемся окне (рис.19) таблицу доступности ресурса добавляем диапазоны дат и максимальное количество единиц ресурса в

данном диапазоне.

Рис.19. Сведения о ресурсе.

Для материального ресурса заносится единица измерения материала и в качестве стандартной ставки используется стоимость за единицу измерения. При вводе ресурсов можно использовать как конкретные, так и универсальные названия. Под *универсальным ресурсом* подразумевается ресурс, обозначающий определенную должность, ей могут соответствовать разные сотрудники. *Конкретный ресурс* относится к определенному сотруднику. По мере разработки проекта универсальные ресурсы заменяются конкретными. К любому ресурсу можно добавить любую дополнительную информацию. Для этого необходимо, выделив ресурс в *Листе ресурсов*, нажать кнопку *Заметки ресурса* на панели инструментов *Стандартная*. В области заметок можно внести свои комментарии или вставить любой файл данных. Для назначения ресурсов перейдите в диаграмму Ганта и выделите задачу, на которую Вы хотите назначить ресурс. На панели инструментов *Стандартная* нажмите кнопку *Назначить ресурсы*. В открывшемся окне *Назначение ресурсов* (Рис.20) выберите ресурс, который вы хотите назначить и нажмите кнопку *Назначить*. После этого в столбце *Единицы* появится выделенный по умолчанию процент рабочего времени, равный либо максимальным единицам данного ресурса (максимум до 100%), либо 100% (максимум 100% или более). Процент может быть изменен на любой другой в пределах максимальных единиц ресурса. Для материальных ресурсов единицами являются количество ресурса. На одну задачу может быть назначено несколько ресурсов.

Рис.20. Окно назначения ресурсов.

Также через окно назначения вы можете снять ресурс с задачи (нажав кнопку *Удалить*), заменить один ресурс на другой (кнопка *Заменить*) или вывести график трудозатрат (кнопка *Графики*). Для задач с фиксированным количеством ресурса и фиксированными трудозатратами добавление новых ресурсов снижает длительность. Если же задача имеет

фиксированную длительность, то добавление новых ресурсов снижает количество единиц для каждого из назначенных ресурсов. Для тех случаев, когда добавление новых ресурсов не должно приводить к изменению длительности, отключают фиксированный объем работ в окне *Сведения о задаче* на вкладке *Дополнительно*. При назначении ресурса задаче, рабочее время ресурса по умолчанию равномерно распределяется по всей длительности задачи. Форма распределения трудозатрат называется профилем загрузки. Для его изменения можно внести свои значения загрузки в табличную часть представления использования ресурса (пункт *Использование ресурса* меню *Вид*). Также можно применить к назначению один из встроенных профилей загрузки. Для этого выбрав назначение, которому необходимо придать определенный профиль, нажимаем кнопку *Сведения о назначении*. На вкладке *Общие* выбирается профиль загрузки из предложенного списка (рис.21).

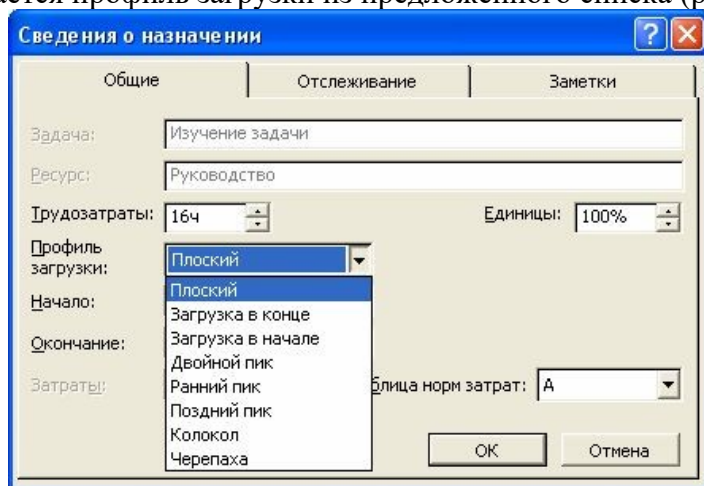


Рис.21. Изменение профиля загрузки.

Контрольные вопросы:

1. Как изменить единицу измерения ресурса?
2. В чем разница между видами начисления затрат?
3. Как изменить календарь, чтобы трудовой ресурс был задействован в проекте один день через каждые два свободных?

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5

Тема: Проверка плана проекта

Цель: проверка проекта с учетом ограничений.

Порядок выполнения

1. Обзор планируемых затрат проекта.
2. Работа с критическими путями и критическими задачами.
3. Балансировка загрузки ресурсов.

Пояснения к выполнению

Для любого проекта наиболее актуальны три группы ограничений – затраты, длительность и использование ресурсов.

Просмотреть затраты на назначения можно, применив таблицу *Затраты* к представлению задач или ресурсов. Для этого в меню *Вид* выбираем команду *Использование задач* или *Использование ресурсов*. После этого также в меню *Вид* выбираем пункт *Таблица: Затраты*. В представлении использования задач отображаются затраты на отдельные назначения, а также общие затраты на каждую задачу (рис.22). В представлении использования ресурсов отображаются затраты на отдельные назначения и общие затраты на каждый ресурс.

	Название задачи	Фиксированные затраты	Начисление фикс. затрат	Общие затраты	Базовые	Откл	Подробно	В	С	Т
1	Развертывание Micros	0,00р.	пропорциональное	0,00р.	0,00р.		Трудозатр			
2	Формирование пре	0,00р.	пропорциональное	0,00р.	0,00р.		Трудозатр			
3	Оценка возможнс	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
4	Определение цел	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
5	Формулировка п	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
6	Определение обл	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
7	Определение осн	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
8	Получение финан	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
9	Формирование п	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
10	Перед началом пла	0,00р.	пропорциональное	0,00р.	0,00р.		Трудозатр			

	Название задачи	Фиксированные затраты	Начисление фикс. затрат	Общие затраты	Базовые	Откл	Подробности	В	С	Т
1	Развертывание Micros	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
2	Формирование пре	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
3	Оценка возможнс	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
4	Определение цел	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
5	Формулировка пр	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
6	Определение обл	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
7	Определение осн	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
8	Получение финан	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
9	Формирование пр	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			
10	Перед началом пла	0,00р.	Пропорциональное	0,00р.	0,00р.		Трудозатр			

Рис.22. Представление использования задач с указанием затрат.

Повременное распределение затрат можно увидеть, добавив сведения о затратах в табличную часть представления использования задач или ресурсов. Для этого, перейдя в представление *Использование задач* или *Использование ресурсов*, необходимо в меню *Формат* выбрать пункт *Подробности/Затраты*. После этого в столбец *Подробности* табличной части представления будет добавлено поле *Затраты*.

Для просмотра данных в другом временном масштабе используются кнопки *Увеличить/Уменьшить* на панели инструментов *Стандартная* (). Для того, чтобы увидеть затраты, связанные с выполнением ресурсами назначенных им задач, необходимо в таблицу ресурсов вставить столбец *Затраты*. Для этого, выбрав представление *Лист ресурсов*, нажимаем на заголовке столбца, справа от которого мы хотим вставить столбец затрат. В меню *Вставка* выбираем команду *Столбец*. В открывшемся окне (рис.23) в списке *Имя поля* выбираем *Затраты*. После закрытия окна в таблице появится поле *Затраты*, в котором будут показаны планируемые затраты на все назначения для каждого ресурса.

Определение столбца

Имя поля: Затраты

Текст заголовка:

Выравнивание заголовка: по центру

Выравнивание данных: по правому краю

Ширина: 10

☒ Перенос заголовка по словам

ОК Отмена Автоширина

Рис.23. Определение столбца.

Ресурсы можно сортировать, группировать и фильтровать по затратам:

- Для сортировки по затратам необходимо, открыв представление *Лист ресурсов*, выбрать пункт меню *Сортировка/по затратам* в меню *Проект*. Пункт меню *Сортировка/по идентификатору* возвращает исходный порядок.
- Для фильтрации необходимо выбрать в меню *Проект* команду *Фильтр/Другие фильтры*. Из открывшегося списка выбираем вариант *Затраты больше чем*. После нажатия на кнопку *Применить* необходимо ввести пороговое значение затрат. Выбрав фильтр *Все ресурсы* возвращаем исходную ситуацию.
- Для группировки необходимо выбрать команду *Группировать по/Стандартной ставке* в меню *Проект*. Для отмены группировки выбираем пункт меню *Группировать по/нет группировки*. Планируемые затраты можно увидеть в отчетах о бюджете и движении денежных средств. Для отображения отчета *Бюджет* необходимо выбрать пункт *Отчеты* в меню *Вид*. Далее выбираем категорию *Затраты* и вид отчета *Бюджет* (рис.24.). В появившемся отчете отображаются фиксированные затраты и общие затраты по задачам.

Отчет о бюджете от 4т 16.08.07
Развертывание Microsoft SharePoint Portal Server

Ид.	Название задачи	Фиксированные затраты	Начисление фикс. затрат	Общие затраты
3	Оценка возможностей сервера Microsoft	0,00р.	Пропорциональное	0,00р.
4	Определение целей организации для р	0,00р.	Пропорциональное	0,00р.
5	Формулировка предварительного анал	0,00р.	Пропорциональное	0,00р.
6	Определение области охвата проекта (	0,00р.	Пропорциональное	0,00р.
7	Определение основных вех	0,00р.	Пропорциональное	0,00р.
8	Получение финансирования от руково,	0,00р.	Пропорциональное	0,00р.
9	Формирование представления и опред	0,00р.	Пропорциональное	0,00р.
12	Определение и замещение должности	0,00р.	Пропорциональное	0,00р.

Ид.	Название задачи	Фиксированные затраты	Начисление фикс. затрат	Общие затраты
3	Оценка возможностей сервера Microsoft	0,00р.	Пропорциональное	0,00р.
4	Определение целей организации для р	0,00р.	Пропорциональное	0,00р.
5	Формулировка предварительного анал	0,00р.	Пропорциональное	0,00р.
6	Определение области охвата проекта (	0,00р.	Пропорциональное	0,00р.
7	Определение основных вех	0,00р.	Пропорциональное	0,00р.
8	Получение финансирования от руково,	0,00р.	Пропорциональное	0,00р.
9	Формирование представления и опред	0,00р.	Пропорциональное	0,00р.
12	Определение и замещение должности	0,00р.	Пропорциональное	0,00р.

Рис.24. Отчет о бюджете.

Отчет *Движение денежных средств*, также выбираемый из категории Затраты, отражает общие планируемые затраты на задачи за недельные периоды (рис. 25).

Движение денежных средств от Чт 16.08.07
Новое предприятие

	29.12.03	05.01.04
Этап 1 - Стратегическое планирование		
Самооценка		
Формирование представления о новом предприятии	800,00р.	
Определение имеющихся умений, навыков, информации и поддержки	800,00р.	
Принятие решения о продолжении		800,00р.
Определение перспектив		
Исследование рынка и конкуренции		

Рис.25. Отчет о движении денежных средств.

Общие затраты на весь проект можно увидеть в окне статистики проекта (рис.26). Для этого необходимо в меню *Проект* выбрать команду *Сведения о проекте*. В открывшемся окне нажимаем на кнопку *Статистика*.

Статистика проекта для 'Новое предприятие'

	Начало	Окончание
Текущее	Чт 01.01.04	Вт 22.06.04
Базовое	НД	НД
Фактическое	НД	НД
Отклонение	0д	0д

	Длительность	Трудозатраты	Затраты
Текущие	124д	1 368ч	56 000,00р.
Базовые	0д?	0ч	0,00р.
Фактические	0д	0ч	0,00р.
Оставшиеся	124д	1 368ч	56 000,00р.

Процент завершения
Длительность: 0% Трудозатраты: 0%

Закрыть

Рис.26. Статистика проекта.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 6

Тема: Корректировка плана проекта.

Цель: Корректировка проекта с учетом ограничений.

Планирование в MS Project осуществляется на основании метода критического пути. Для этого анализируются все последовательности связанных задач в проекте и определяется, какая из последовательностей имеет наименьший временной резерв, она и обозначается как критический путь. В норме задачи, лежащие на критическом пути, имеют нулевой временной резерв. Однако, можно поменять определение критической задачи, используемое программой. Для этого необходимо выбрать пункт *Параметры* в меню *Сервис* и в открывшемся окне перейти на вкладку *Расчет* (рис.27). Значение максимального временного резерва критической задачи устанавливается в поле *Считать критическими задачи, имеющие резерв не более*. Также на этой вкладке можно настроить режим использования

нескольких критических путей. В этом случае Microsoft Project изменяет расчет критического пути таким образом, что любая задача, не имеющая последователя, становится критической. Это значит, что ее дата позднего окончания устанавливается равной дате раннего окончания и временной резерв оказывается равным нулю. Любая последовательность задач, предшествующих данной, становится критическим путем.

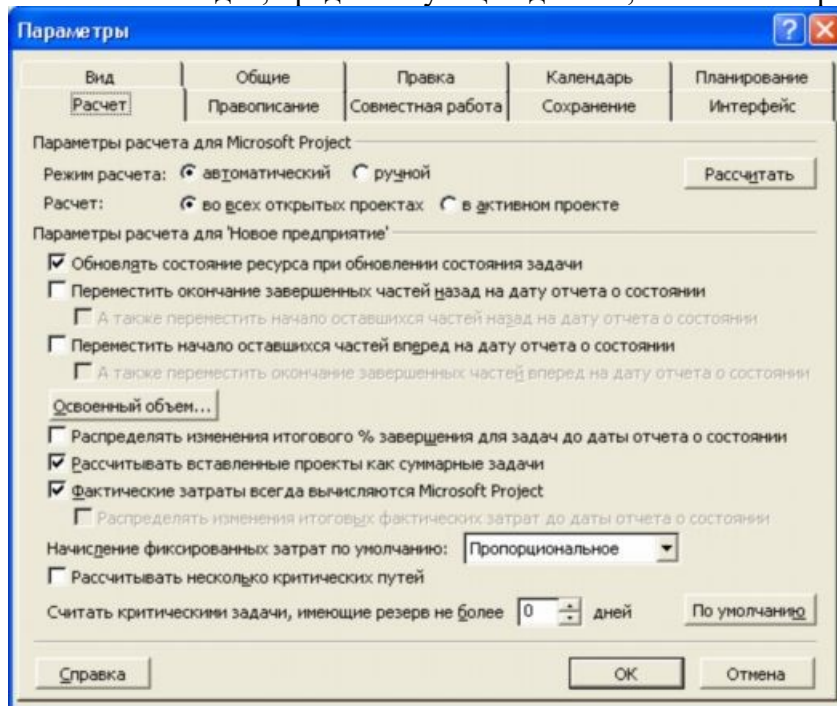


Рис.27. Параметры расчета.

Для того, чтобы узнать величину свободного и полного временного резерва необходимо, открыв диаграмму Ганта, выбрать пункт *Таблица: Календарный план* из меню *Вид*. После этого в табличной части появятся поля *Свободный временной резерв* и *Общий временной резерв*. Для того, чтобы увидеть критический путь на диаграмме Ганта, необходимо выбрать пункт *Диаграмма Ганта с отслеживанием* в меню *Вид*. На этой диаграмме критический путь отображается красным цветом в графической части представления(рис.28).



Рис.28. Диаграмма Ганта с отслеживанием.

В проекте, ограниченном по ресурсам, необходимо обеспечить максимальное использование всех ресурсов, назначение их на правильные задачи и недопущение превышения доступности. Ресурсы с превышением доступности обозначаются в *Листе ресурсов* красным цветом. Чтобы увидеть, на какие задачи назначены ресурсы с превышением доступности необходимо в добавить в представление задач (например, диаграмму Ганта) столбец *Превышение доступности*. Для этого необходимо, перейдя в представление задач, выбрать пункт *Столбец* из меню *Вставка*. В открывшемся окне в списке *Имя поля* выберите *Превышение доступности*. Если у задачи есть перегруженные ресурсы, то в данном столбце будет значение *Да*. Чтобы увидеть какое количество работы назначено ресурсу в течение определенного времени можно использовать *График ресурсов*. Для этого необходимо выбрать представление *График ресурсов* в меню *Вид*. По умолчанию показываються пиковые единицы, однако можно выбрать например трудозатраты или процент загрузки. Для этого нажимаем в области окна вида правой кнопкой мыши и выбираем нужный режим из контекстного меню. Более подробно сведения об использовании ресурса отображаются в

Форме ресурса (рис.29). Для того, чтобы ее увидеть, необходимо открыть представление Лист ресурсов и затем выбрать в меню *Окно* команду *Разделить*. После этого в нижней части окна появится форма ресурсов, в которой будет показана информация о выбранном ресурсе.

	И	Название ресурса	Тип	Единицы измерения материалов	Краткое название	Группа	Макс. единиц	Стандартная ставка	Ставка сверхурочных	Зв
1	!	Ответственный за пер	Трудовой		О		100%	0,00р./ч	0,00р./ч	
2	!	Руководитель офиса	Трудовой		Р		100%	0,00р./ч	0,00р./ч	
3	!	Администратор систем	Трудовой		А		100%	0,00р./ч	0,00р./ч	
4	!	Руководитель сетевой	Трудовой		Р		100%	0,00р./ч	0,00р./ч	
5	!	Администратор почтов	Трудовой		А		100%	0,00р./ч	0,00р./ч	
6	!	Специалист по охране	Трудовой		С		100%	0,00р./ч	0,00р./ч	
7		Бухгалтер	Трудовой		Б		100%	0,00р./ч	0,00р./ч	
8		Местная телефонная ком	Трудовой		М		100%	0,00р./ч	0,00р./ч	
9	!	Служба уборки	Трудовой		С		100%	0,00р./ч	0,00р./ч	
10		Веб-мастер	Трудовой		В		100%	0,00р./ч	0,00р./ч	
11	!	Перевозчики	Трудовой		П		100%	0,00р./ч	0,00р./ч	

Название: Администратор систем связи
Краткое назв.: А
Макс. ед: 100%

Предыдущий
Следующий

Затраты

Стандартная ставка: 0,00р./ч
На использование: 0,00р.

Ставка сверхурочных: 0,00р./ч
Начисление: Пропорциональное

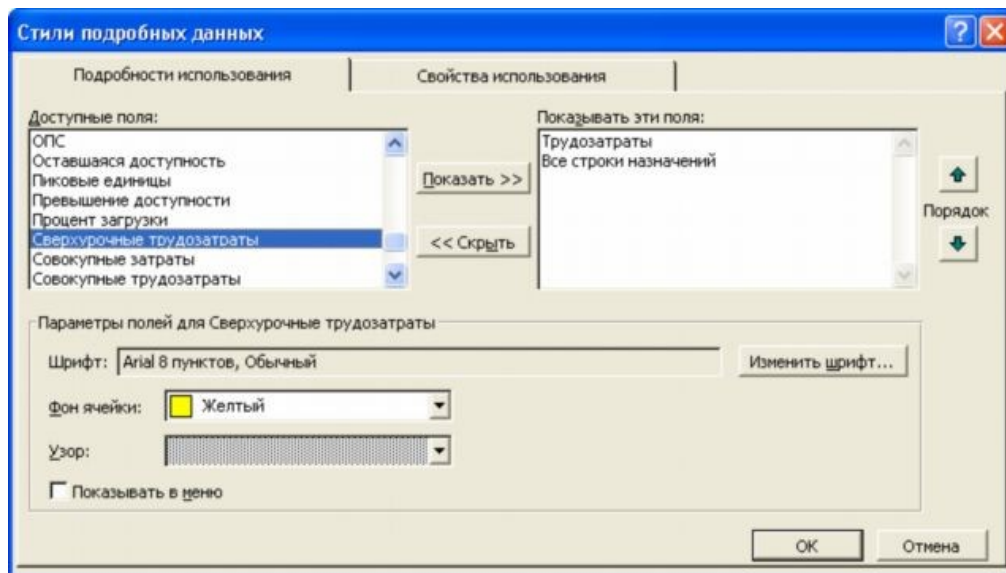
Баз. календарь: Стандартный
Группа:
Код:

Проект	Ид.	Название задачи	Трудозатраты	Выравн. задержка	Задержка	Начало	Окончание
Переезд о	14	Оценка требований к телеф	8ч		0д	Пн 08.03.04	Пн 08.03.04
Переезд о	15	Заказ нового номера телефо	8ч		0д	Пн 08.03.04	Пн 08.03.04
Переезд о	16	Заказ нового номера факса	8ч		0д	Пн 08.03.04	Пн 08.03.04
Переезд о	34	Заказ телефонной системы	8ч		0д	Чт 20.05.04	Чт 20.05.04
Переезд о	35	Заказ телефонных линий	8ч		0д	Пт 21.05.04	Пт 21.05.04
Переезд о	55	Подключение международн	8ч		0д	Чт 27.05.04	Чт 27.05.04

Рис.29. Форма ресурсов.

При перегрузке ресурсов можно:

- Заменить перегруженный ресурс на другой. Для этого необходимо, выбрав на диаграмме Ганта задачу с перегруженными ресурсами, нажать кнопку Назначение ресурсов () на панели инструментов Стандартная. В диалоговом окне Назначение ресурсов выбрать заменяемый ресурс и нажать кнопку *Заменить*. В окне Замена ресурсов выбрать тот ресурс, на который необходимо провести замену;
- Назначить дополнительный ресурс;
- Отложить задачу или назначение до тех пор, пока у ресурса не появится свободное время. Для этого используется выравнивающая задержка и задержка назначения. *Выравнивающая задержка* – количество времени которое должно пройти от планируемой даты начала задачи до момента фактического начала выполнения. При этом задерживаются все назначения по данной задаче. Для добавления выравнивающей задержки необходимо, перейдя в представление *Диаграмма Ганта с отслеживанием*, выбрать из меню *Вид* пункт *Таблица/Другие таблицы*. Из предлагаемого списка выбираем таблицу *Задержка* и нажимаем *Применить*. После этого выравнивающая задержка вводится в соответствующее поле. *Задержка назначения* – количество времени, которое должно пройти от планируемой даты начала выполнения задачи до планируемой даты начала назначения. Для добавления задержки назначения необходимо вставить в табличную часть диаграммы Ганта с отслеживанием столбец *Задержка назначения*;
- Назначить сверхурочные. Для этого переходим в представление *Использование ресурсов* и добавляем в табличную часть столбец *Сверхурочные трудозатраты*. Для отображения сверхурочных затрат по временным периодам необходимо в меню *Формат* выбрать пункт *Стили подробных данных*. В открывшемся окне (рис.30) в списке *Доступные поля* выбираем *Сверхурочные трудозатраты* и нажимаем на кнопке *показать*. После закрытия окна сверхурочные затраты появляются в обеих частях



представления.

Рис.30. Добавление полей подробных данных.

Сверхурочные затраты можно добавить только в табличную часть, изменить их автоматическое распределение по временным периодам нельзя.

· прервать задачу. Для того, чтобы прервать выполнение задачи и вернуться к ней с определенной даты, необходимо перейти в представление *Диаграмма Ганта*, и нажать кнопку *Прервать задачу*. После этого в графической части необходимо выбрать прерываемую задачу и указать дату, где должно произойти прерывание. Вторую часть отрезка перетаскиваем к той дате, где должно произойти прерывание.

Контрольные вопросы:

1. Что нам дает группировка ресурсов?
2. Как взаимосвязаны между собой свободный и полный временной резерв задачи?
3. В каких случаях и какие методы избавления от перегрузки ресурсов используются?

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 7

Тема: Основы работы в BPwin 4.1

Цель: познакомиться с основными элементами рабочего интерфейса BPwin

4.1; освоить технологию создания новой модели; научиться редактировать построенную диаграмму.

Теоретические сведения

Для упрощения процесса построения модели бизнес-процессов в IDEF0, можно использовать CASE-средство BPwin, которое позволяет наглядно представить функциональную декомпозицию системы.

AllFusion Process Modeler 4.1 (BPwin) является инструментальным средством, полностью поддерживающим стандарт IDEF0, который был принят в начале 90-х годов в США на основе SADT.

Основная идея методологии SADT - построение **древовидной функциональной модели** предприятия.

Сначала функциональность предприятия описывается в целом, без подробностей. Такое описание называется **контекстной диаграммой** (рис. 1.1.).

Взаимодействие с окружающим миром описывается в терминах **входа** (данные или объекты, потребляемые или изменяемые функцией), **выхода** (основной результат деятельности функции, конечный продукт), **управления** (стратегии и процедуры, которыми руководствуется функция) и **механизмов** (необходимые ресурсы).

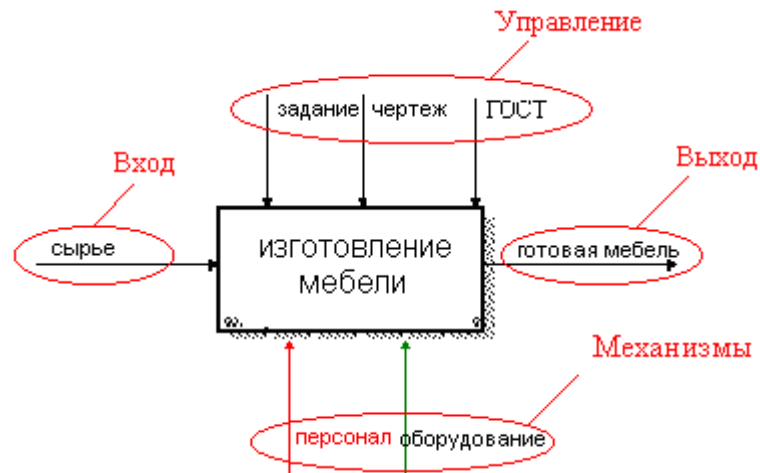


Рис. 1.1. Пример диаграммы IDEF0

Кроме того, при создании контекстной диаграммы формулируются *цель моделирования, область* (описания того, что будет рассматриваться как компонент системы, а что как внешнее воздействие) и *точка зрения* (позиция, с которой будет строиться модель). Обычно в качестве точки зрения выбирается точка зрения лица или объекта, ответственных за работу моделируемой системы в целом.

Общая функция разбивается на крупные подфункции. Этот процесс называется **функциональной декомпозицией**.

Затем каждая подфункция декомпозируется на более мелкие - и так далее до достижения необходимой детализации описания. Таким образом, формируется диаграмма IDEF0.

Этапы построения модели

1. Определение основного бизнес-процесса.
2. Построение контекстной диаграммы.
3. Построение диаграммы процесса верхнего уровня.
4. Функциональная декомпозиция каждого процесса, с помощью детализирующих диаграмм.

Практическое задание

Создание диаграммы А-0

Построение модели бизнес-процесса рассмотрим на примере работы мебельной фабрики. Во время проведения обследования предприятия были выявлены её целевые задачи, функциональные деятельности каждого из подразделений предприятия и функциональные взаимодействия между ними; информационные потоки внутри подразделений и между ними; внешние по отношению к предприятию объекты и внешние информационные воздействия, а так же нормативно-справочная документация, данные по имеющимся на предприятии средствам и системам автоматизации.

Целевые функции мебельной фабрики:

- переработка сырья;
- изготовление деталей для мебели;
- сборка изделия;
- контроль качества.

Нормативные документы мебельной фабрики:

- чертежи (деталей, сборочный);
- нормы по переработке сырья;
- стандарты качества;
- производственные инструкции;

- инструкции по технике безопасности.

Подразделения предприятия:

- цех по обработке сырья и бракованных изделий;
- цех по изготовлению деталей;
- сборочный цех;
- отдел проверки качества изделия.

Основным сырьем для изготовления мебели является *дерево*.

Определим основной бизнес-процесс, используя выявленные целевые функции. Так как основное предназначение мебельной фабрики состоит в том, чтобы изготавливать мебель, значит, основным бизнес-процессом является **ИЗГОТОВЛЕНИЕ МЕБЕЛИ**.

Создадим контекстную диаграмму. В данном случае она представляет собой самое общее описание системы. Напомню, что в модели может быть только одна контекстная диаграмма.

Для построения контекстной диаграммы нам необходимо определить входную информацию (данные или материальные ресурсы), которая преобразуется в процессе для получения результата; выходную информацию - готовый результат; управление, которое влияет на процесс, но не преобразуется процессом.; механизмы, которые выполняют процесс.

Для контекстного процесса **ИЗГОТОВЛЕНИЕ МЕБЕЛИ** определим необходимую информацию:

- **ВХОД** - сырьё;
- **УПРАВЛЕНИЕ** – чертежи, производственные инструкции, инструкции по технике безопасности (нормативные документы);
- **МЕХАНИЗМЫ** – персонал, производственное оборудование;
- **ВЫХОД** – готовая мебель.



Задание 1. Запуск программы и открытие проекта.

Создание новой модели.

1. Запустите **BPwin**.
2. Выберите режим работы: создать новую модель (**Create model**).
3. В поле **Name** введите имя модели: **Изготовление мебели**.
4. Из группы **Type** выберите тип диаграммы **Business Process (IDEF0)** (рис. 1.2.).

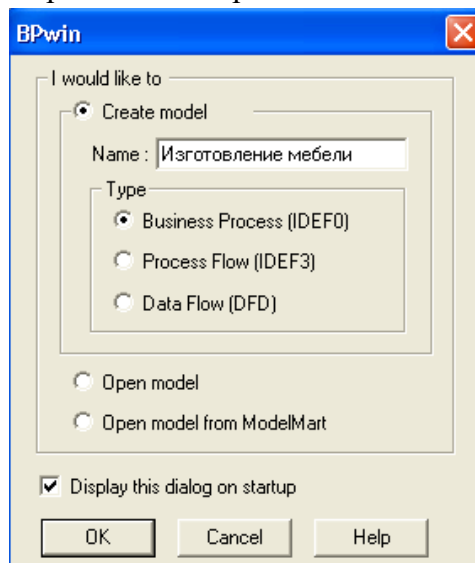


Рис. 1.2. Диалоговое окно создания модели

5. Нажмите **OK**.

После щелчка на кнопке **OK** появляется диалог **Properties for New Models** (рис. 1.3.).

6. Укажите свою фамилию и инициалы.

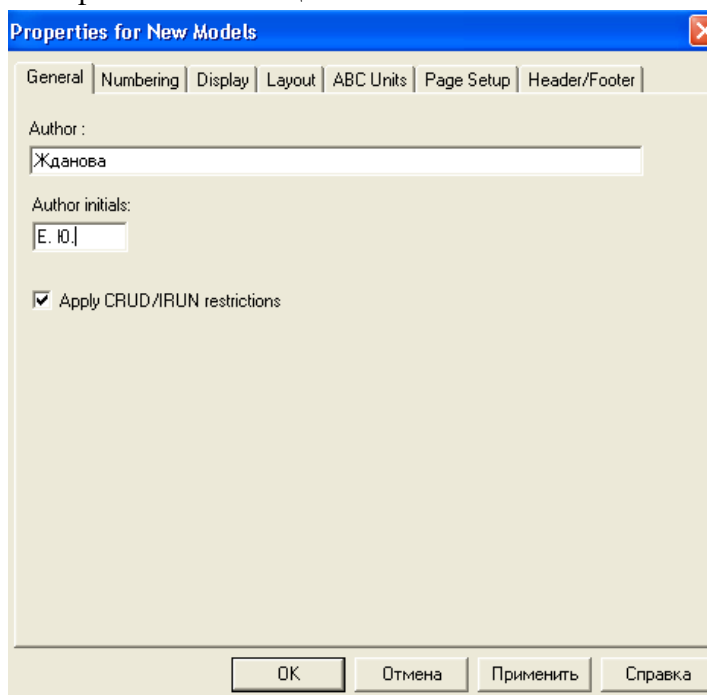


Рис. 1.3. Диалоговое окно *Properties for New Models*

После этого вы увидите стандартное окно BPwin (рис. 1.4.):

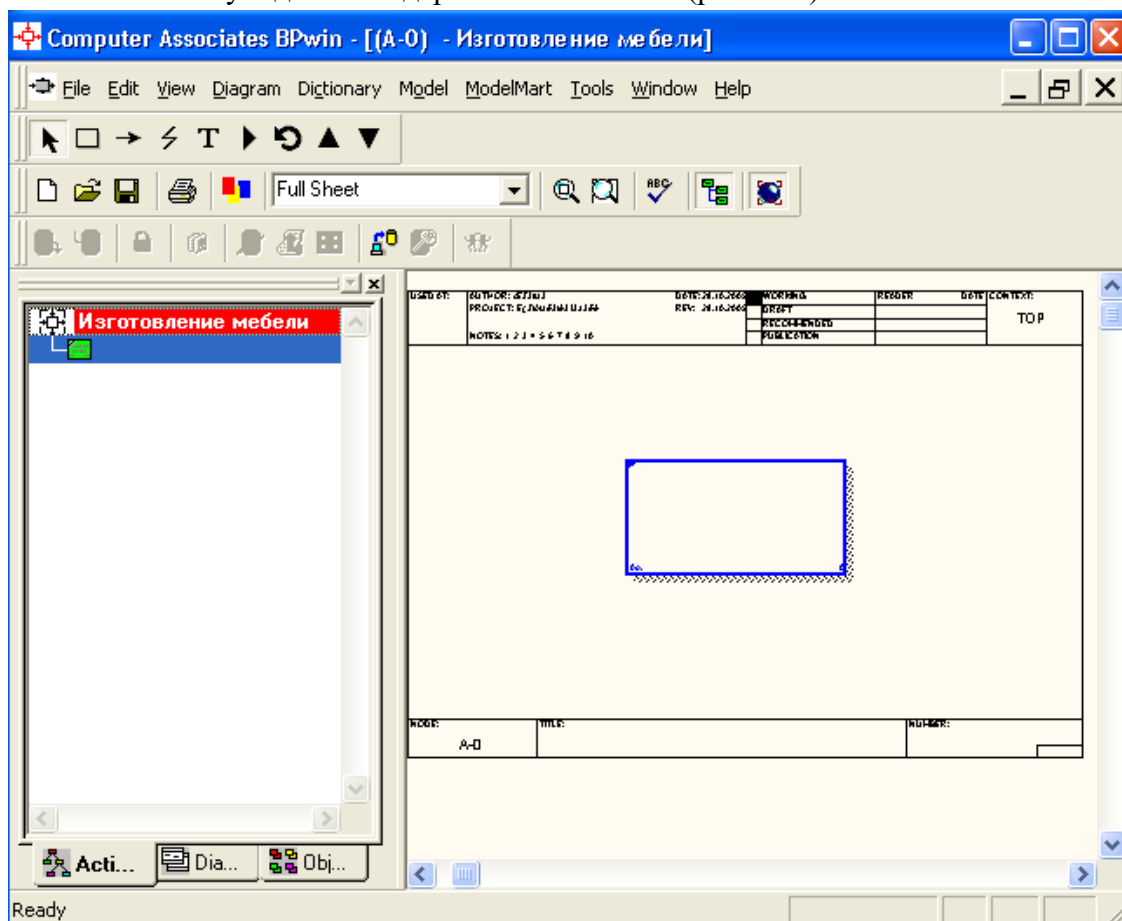


Рис. 1.4. Стандартное окно *BPwin*

Основные инструменты BPwin

Все основные действия с диаграммами, такие как создание, редактирование и т.д., можно выполнить с помощью главного меню, либо контекстно-зависимого меню (меню,

появляющееся при нажатии правой кнопки мыши). Принципы работы с меню являются стандартными для среды Windows: объект сначала делается активным, затем над ним осуществляются необходимые действия.

На основной панели инструментов расположены элементы управления, в основном знакомые по другим Windows-интерфейсам (рис. 1.5.):



Рис. 1.5. Элементы управления BPwin

Функциональность панели инструментов доступна из основного меню BPwin (табл. 1).

Таблица 1.1. Элементы управления основной панели инструментов BPwin

Элемент управления	Описание	Соответствующий пункт меню
	Создать новую модель	File/New
	Открыть модель	File/Open
	Сохранить модель	File/Save
	Напечатать модель	File/Print
	Вызвать генератор отчетов Report Builder	Tools/Report Builder
	Выбор масштаба	View/Zoom
	Масштабирование	View/Zoom
	Проверка правописания	Tools/Spelling
	Включение и выключение навигатора модели Model Explorer	View/Model Explorer
	Включение и выключение дополнительной панели инструментов работы с ModelMart	ModelMart

На основной панели инструментов (либо в любом желаемом месте экрана) расположены инструменты редактора BPwin для IDEF0-диаграмм (рис. 1.6.).



Рис. 1.6. Панель инструментов редактора BPwin

Таблица 1.2. Инструменты редактора BPwin

Элемент редактирования	Описание	Название элемента
	Выбор и определение позиции объектов, добавленных в диаграмму.	Pointer Tool
	Установка блоков в диаграмме.	Activity Box Tool
	Установка дуги в диаграмме.	Arrow Tool
	Создание тильды (squiggle), которая соединяет дугу с ее названием.	Squiggle Tool
	Создание текстовых блоков.	Text Block Tool
	Отображение следующей диаграммы того же уровня	Go to Sibling Diagram
	Открытие диалогового окна Diagram Dictionary Editor, где можно перейти на какую-либо диаграмму или создать новую диаграмму.	Diagram Dictionary Editor
	Переход на родительскую диаграмму.	Go to Parent Diagram
	Отображение диаграммы потомка или разложение выделенного блока на диаграмму потомка	Go to Child Diagram



Задание 2. Работа с блоком.

Построение контекстной диаграммы процесса ИЗГОТОВЛЕНИЕ МЕБЕЛИ.

Для ввода имени блока необходимо:

1. Щелкнуть правой клавишей мыши по блоку.
2. Выбрать команду **Name**.
3. В диалоговом окне ввести название «Изготовление мебели» (рис.1.7.).

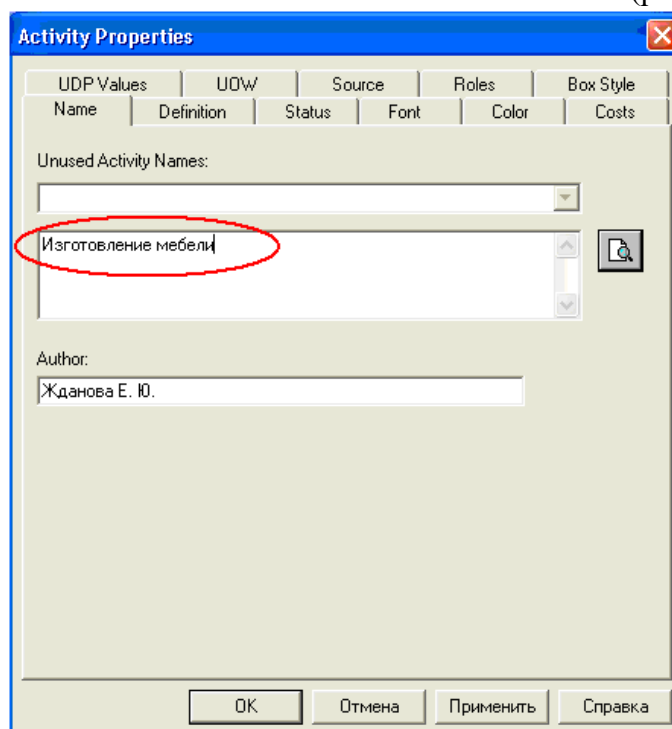


Рис. 1.7. Диалоговое окно BPwin

4. Для того чтобы текст стал понятен, в контекстном меню выберите пункт **Font** (рис. 1.8.):

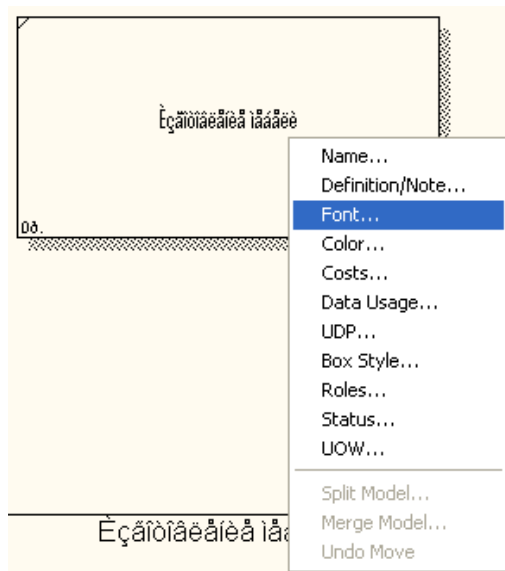


Рис. 1.8. Контекстно-зависимое меню

5. В диалоговом окне **Activity Properties** в нижней части вкладки **Font** установите флажки в опциях **Apply setting to**, позволяющих изменить шрифт для всех работ на текущей диаграмме, в модели, и в группе **Global**, позволяющей изменить шрифт одновременно для всех объектов модели, в опции **Script** выберите «кириллический».

6. Установите шрифт **Arial Unicode MS**, курсив, 16 пт (рис. 1.9.).

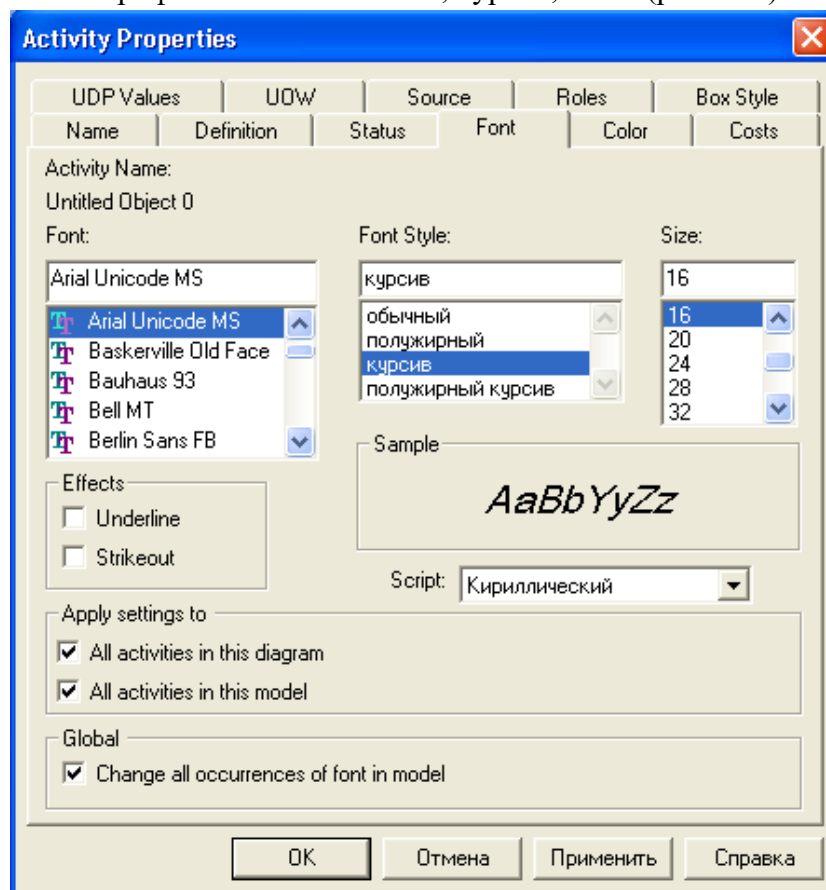


Рис. 1.9. Вкладка Font диалога Activity Properties

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 3. Построение дуг.

Для построения дуги **управления** необходимо:

1. Выбрать на панели инструментов кнопку .

2. Подвести курсор к верхнему краю окна построения диаграммы до появления черной полосы и произвести щелчок левой кнопкой мыши по этой полосе (рис. 1.10.).



Рис. 1.10. Начало построения дуги

3. Подвести курсор мыши к верхней стороне блока до образования темного треугольника и щелкнуть левой кнопкой мыши (рис. 1.11.).

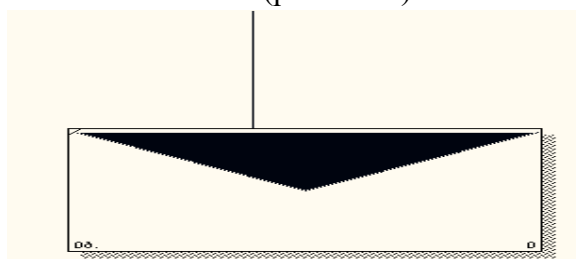


Рис. 1.11. Построение дуги управления

Построение дуг **входа** и **механизмов** производится аналогичным образом.

4. Постройте дуги **входа** и **механизмов**.

Для построения дуги **выхода** выполняются те же действия, но в противоположном порядке: от правой стороны блока к правой стороне окна построения диаграмм.

5. Постройте дугу **выхода**.



Задание 4. Идентификация дуги управления.

1. Выберите на панели редактирования кнопку
2. Щелкните правой кнопкой мыши по дуге.
3. Выберите команду **Name** (рис. 1.12.).

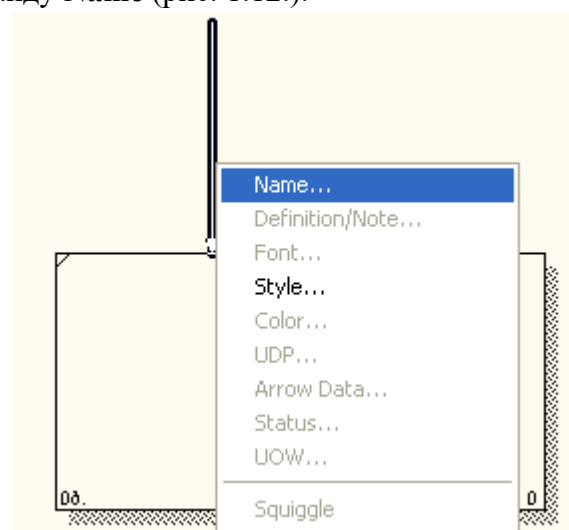


Рис. 1.12. Контекстно-зависимое меню

4. В диалоговом окне введите название дуги: «Нормативная документация» (рис. 1.13.).

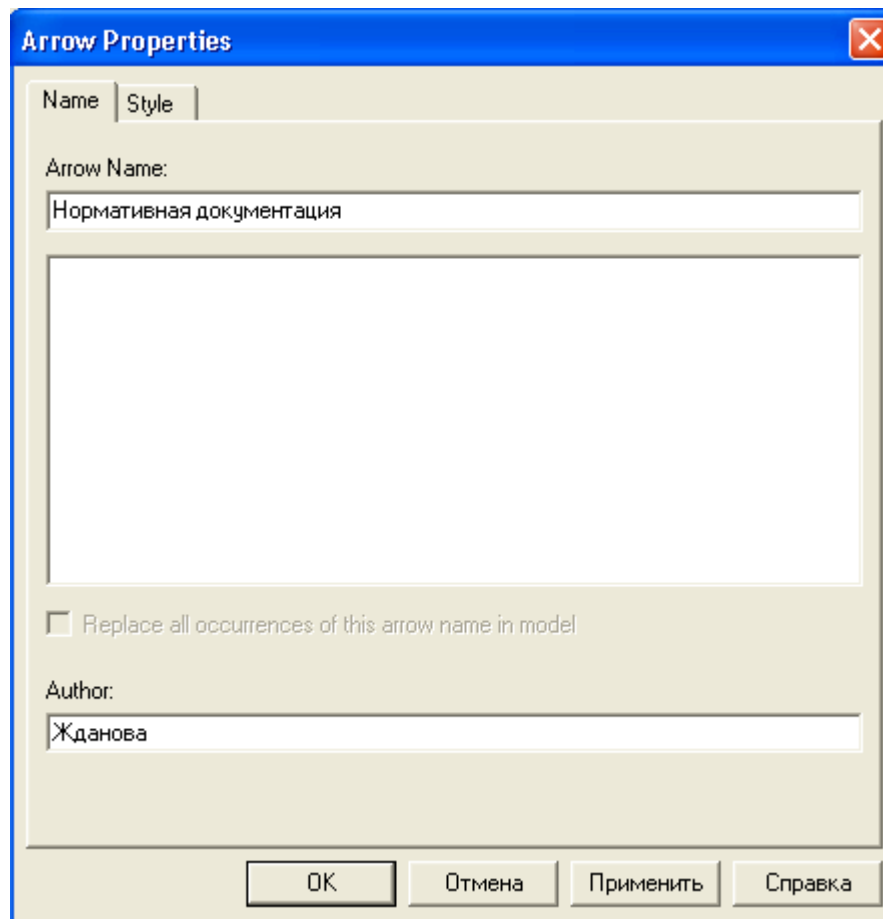


Рис. 1.13. Диалоговое окно Arrow Properties

5. Для того чтобы текст для дуги стал понятен (рис 1.14.), выберите меню **Model - Default Fonts**.

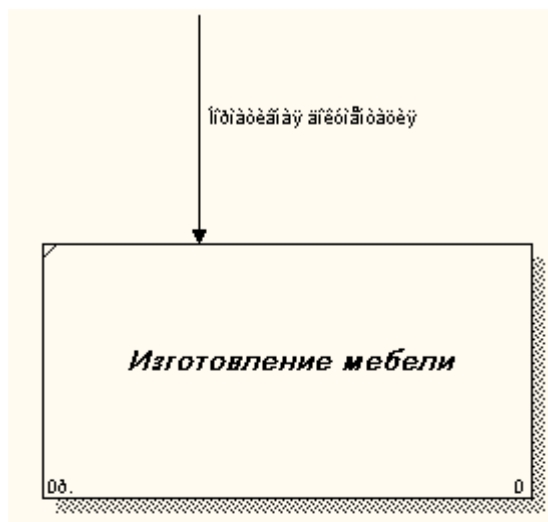


Рис. 1.14. Фрагмент диаграммы

ВРwin позволяет установить шрифт по умолчанию для объектов определенного типа (например, дуг) на диаграммах и в отчетах. После выбора меню **Model - Default Fonts** появляется каскадное меню, каждый пункт которого служит для установки шрифтов для определенного типа объектов (рис. 1.15.):

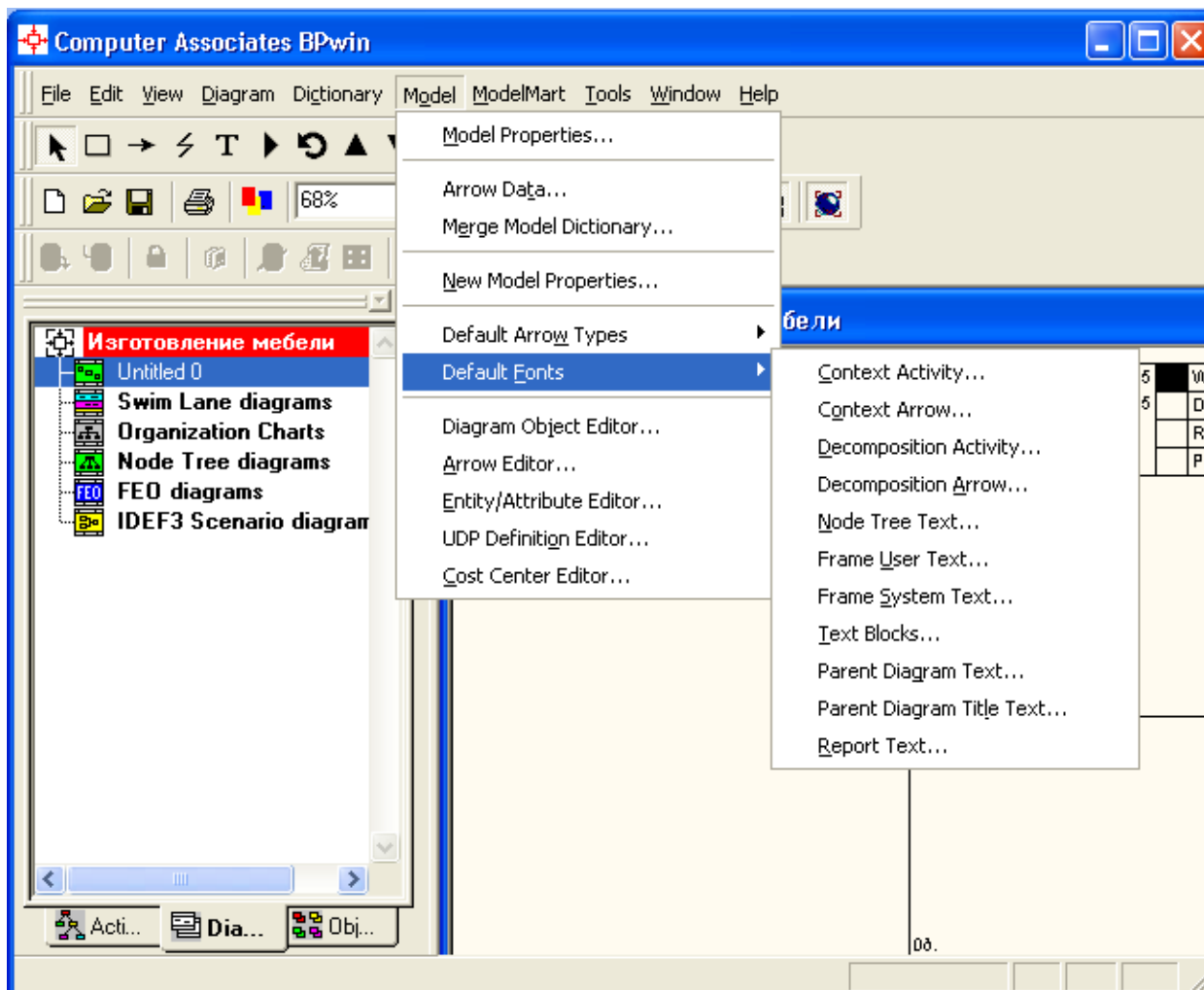


Рис. 1.15. Диалог Model - Default Fonts

1. **Context Activity** - работа на контекстной диаграмме;
2. **Context Arrow**- стрелки на контекстной диаграмме;
3. **Decomposition Activity** - работы на диаграмме декомпозиции;
4. **Decomposition Arrow** - стрелки на диаграмме декомпозиции;
5. **Node Tree Text** - текст на диаграмме дерева узлов;
6. **Frame User Text** - текст, вносимый пользователем в каркасе диаграмм;
7. **Frame System Text** - системный текст в каркасе диаграмм;
8. **Text Blocks** - текстовые блоки;
9. **Parent Diagram Text** - текст родительской диаграммы;
10. **Parent Diagram Title Text** - текст заголовка родительской диаграммы;
11. **Report Text** - текст отчетов.

6. В диалоговом окне **Default Context Arrow Name Text Font** в нижней части установите флажок в опции **Change all occurrences**, позволяющей изменить шрифт для названий всех дуг на текущей диаграмме, в опции **Script** выберите «**кириллический**».

7. Установите шрифт **Arial Unicode MS**, курсив, 14 пт. (рис. 1.16.).

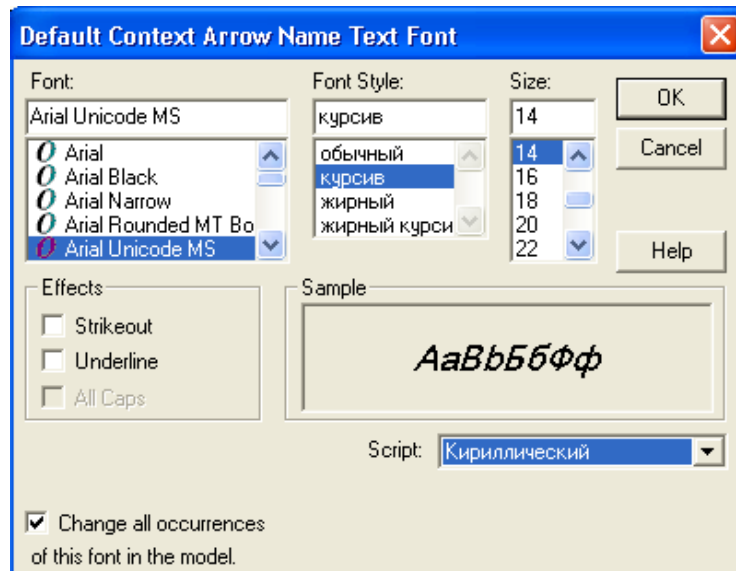


Рис. 1.16. Диалоговое окно Default Context Arrow Name Text Font

Вот что должно у вас получиться (рис. 1.17.).



Рис. 1.17. Фрагмент диаграммы



Задание 5. Работа с блоком.

Самостоятельно постройте дуги:

- ВХОДА: «Сырьё»;
- МЕХАНИЗМА: «Персонал», «Производственное оборудование»;
- ВЫХОДА: «Готовая мебель».



После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 6. Установление тильды.

Название дуги является независимым объектом, который можно перемещать относительно дуги. Текст может располагаться по отношению к дуге в свободной форме, либо соединяться с дугой символом тильды.

Чтобы установить тильду следует:

1. На панели инструментов нажать кнопку  ;
2. Щелкнуть левой кнопкой мыши по тексту, а затем по дуге (рис. 1.18.);

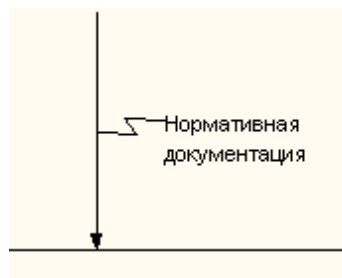


Рис. 1.18. Установка тильды

3. Можно также использовать команду контекстно-зависимого меню **Squiggle** (рис. 1.19.).

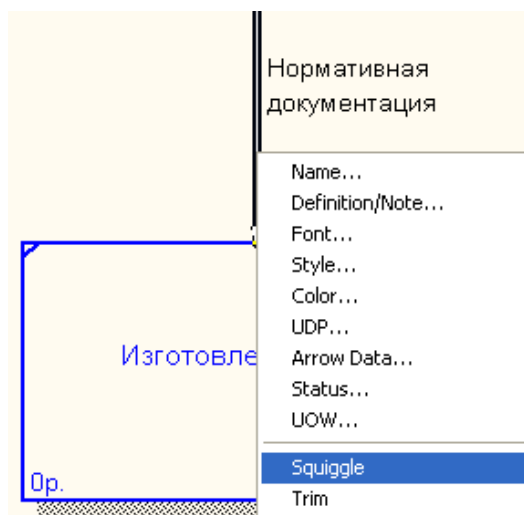


Рис. 1.19. Контекстно-зависимое меню

Дуга представляет собой совокупность отдельных графических объектов: прямые участки, изогнутые участки, изображение наконечника. Отдельные элементы можно передвигать независимо друг от друга, меняя форму дуги, также дугу можно перемещать как единый неделимый элемент.

4. Установите тильду к остальным дугам и их названиям.



Задание 7. Изменение цвета текста, фона блока, цвета и стиля дуг.

1. Для изменения цвета текста выполните команду контекстно-зависимого меню **Color** (рис. 1.20.):

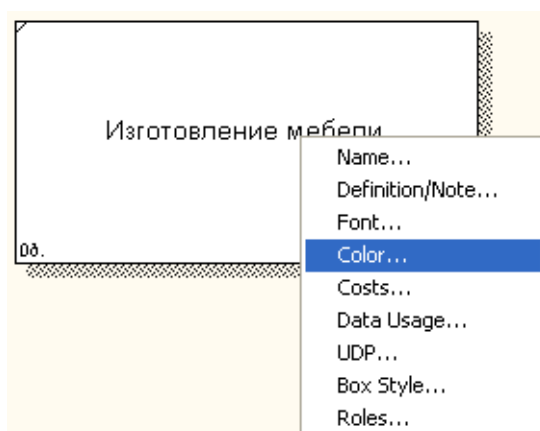


Рис. 1.20. Контекстно-зависимое меню

2. Выберите цвет и нажмите кнопку **OK** (рис. 1.21.).

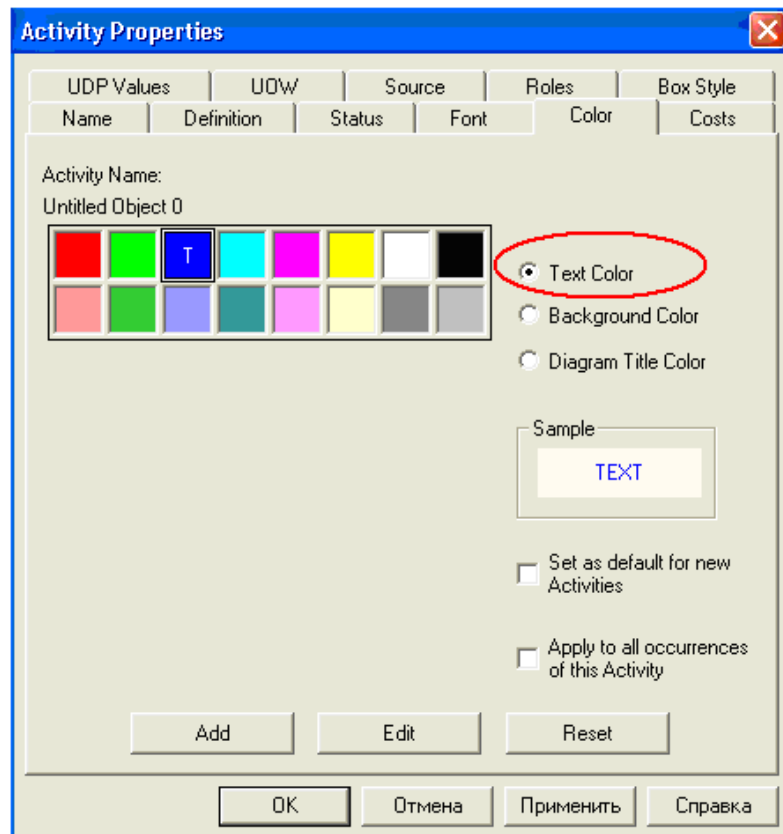


Рис. 1.21. Диалоговое окно выбора цвета текста и стрелок

3. Для изменения фона блока выберите **Background Color** и цвет (рис. 1.22.):

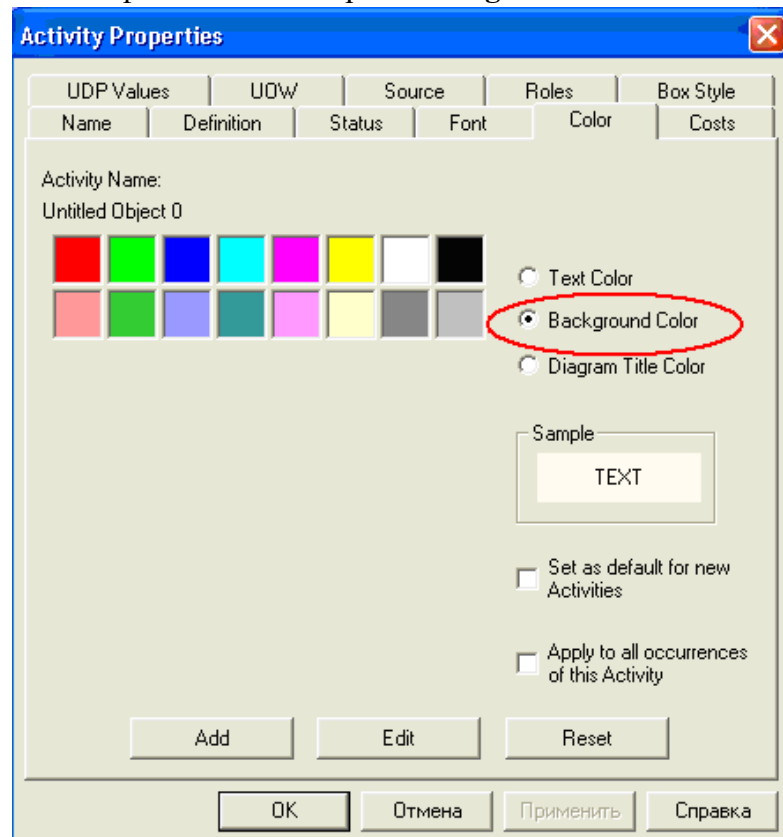


Рис. 1.22. Вкладка Color диалога Activity Properties

4. Для изменения стиля дуги выберите в контекстно-зависимом меню команду **Style** (рис. 1.23.):

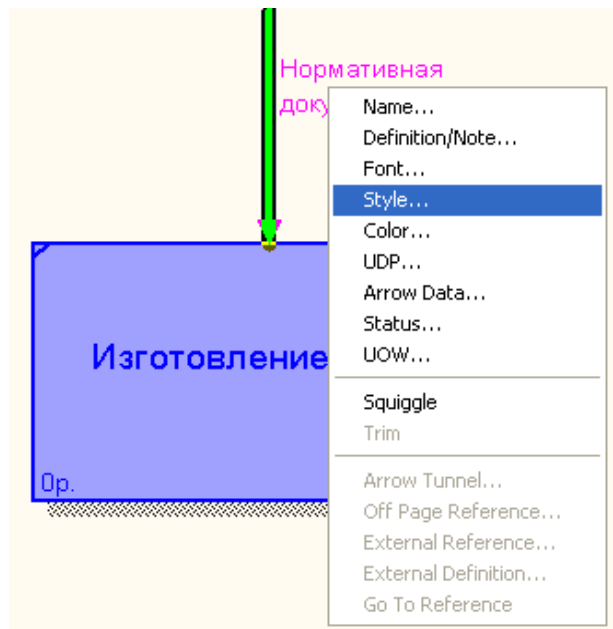


Рис. 1.23. Контекстно-зависимое меню

5. В диалоговом окне укажите тип и стиль дуги, нажмите на кнопку ОК (рис. 1.24.).

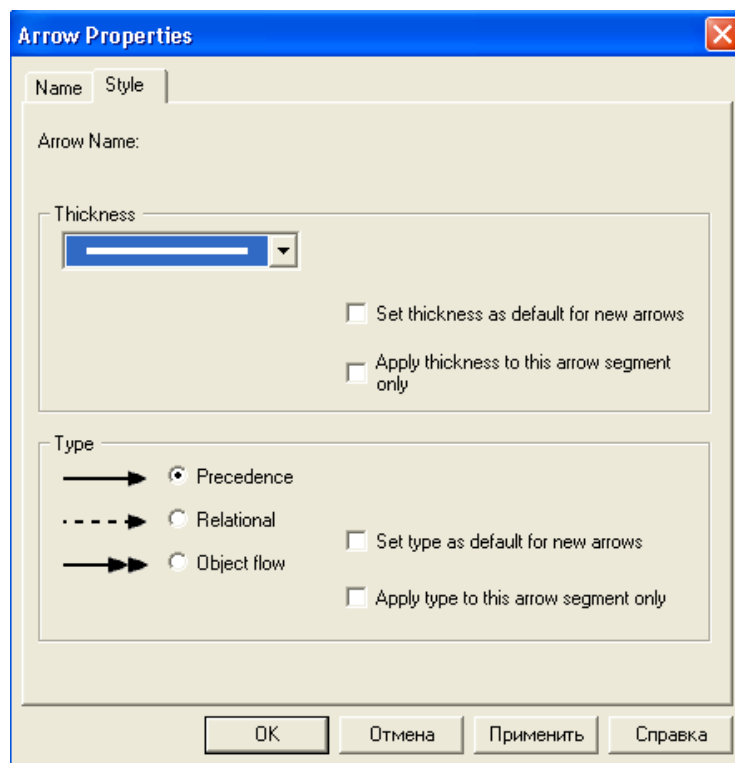


Рис. 1.24. Диалоговое окно Arrow Properties

Вот что должно у вас получиться (рис. 1.25.).

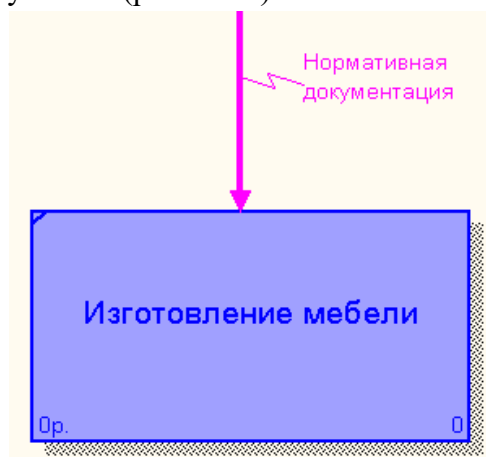


Рис. 1.25. Пример форматирования элементов диаграммы

Удаление блоков, дуг или текста.

Для удаления блока и дуги или текста необходимо их активизировать щелчком левой кнопки мыши и нажать клавишу Delete, а затем подтвердить намерения по поводу удаления.



Задание 8. Форматирование диаграммы.

1. Самостоятельно произведите форматирование всех элементов диаграммы, опираясь на данные, приведенные в таблице 1.3.

Таблица 1.3. Таблица редактирования

Объект	Действие	Порядок выполнения
Блок	Изменение размера	Для изменения высоты перетащить мышью верхнюю или нижнюю границу блока, аналогично меняется размер по горизонтали
Текст	Стандартные способы форматирования	Правой клавишей мышки щелкнуть по тексту, выбрать пункт контекстно-зависимого меню Font Editor .
Дуга	Изменение стиля, цвета, размера	Щелкнуть правой клавишей мышки по дуге и выбрать соответствующий пункт: Style Editor , Color Editor или Trim
Удаление блоков, дуг или текста: активизировать щелчком левой кнопки мышки необходимый объект и нажать Delete , а затем подтвердить запрос на удаление.		

После выполнения задания у вас должна получиться следующая контекстная диаграмма (рис. 1.26.):

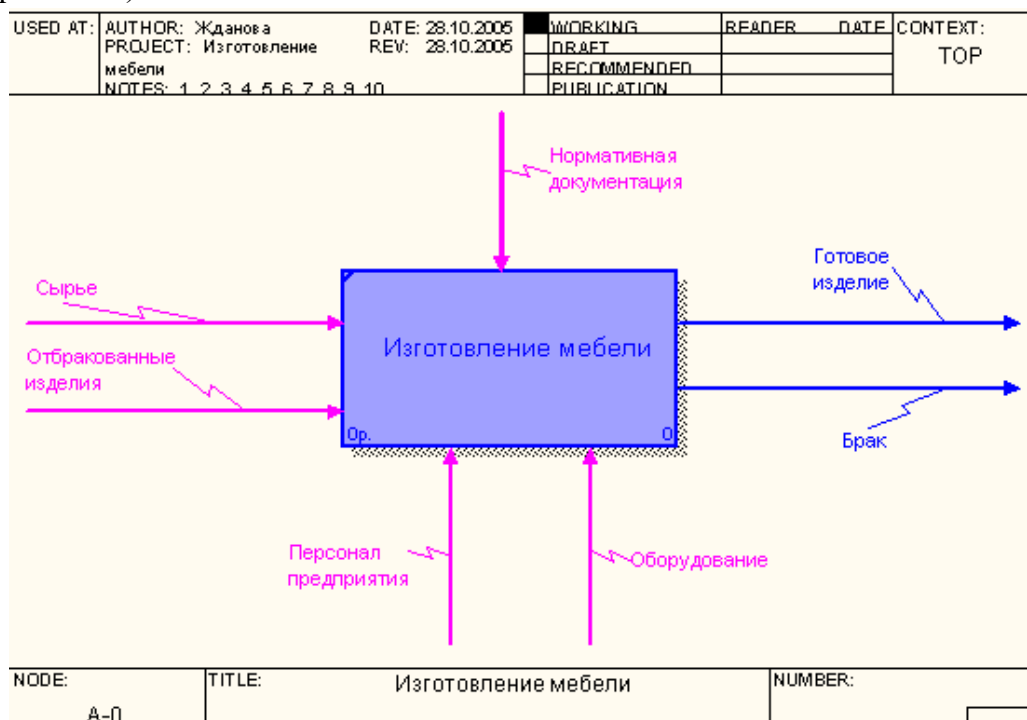


Рис. 1.26. Контекстная диаграмма процесса «Изготовление мебели»



После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 9. Сохранение полученной диаграммы.

Сохраните полученную диаграмму.

1. Создайте папку, назовите её своей фамилией и в неё сохраняйте свои работы.
2. В меню **File** выберите команду **Save as**.
3. Укажите путь к своей папке и имя файла **Lab1.bp1** (рис. 1.27.).

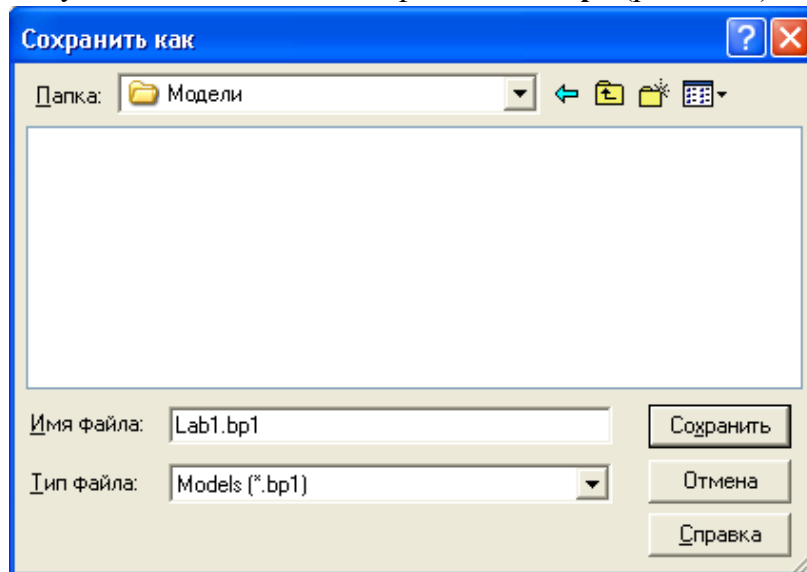


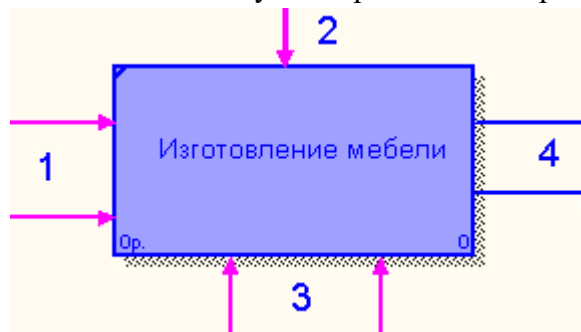
Рис. 1.27. Диалоговое окно сохранения документа.

4. Нажмите на кнопку СОХРАНИТЬ.

Контрольные вопросы

Ответьте на контрольные вопросы:

1. Перечислите основные возможности BPwin.
2. Охарактеризуйте основные элементы рабочего интерфейса BPwin.
3. Какую методологию поддерживает BPwin?
4. Укажите назначение каждой из дуг изображенных на рисунке.



5. Назовите основные этапы построения модели.
6. Какой процесс можно назвать функциональной декомпозицией?
7. Перечислите элементы контекстной диаграммы.
8. При помощи какого инструмента строятся дуги на диаграмме?



Тестовое задание

1. Что представляет собой модель бизнес-процессов:
 - a) комплекс диаграмм, каждая из которых описывает отдельный бизнес-процесс;
 - b) иерархию диаграмм, каждая из которых описывает отдельный бизнес-процесс;
 - c) случайный набор диаграмм, каждая из которых описывает отдельный бизнес-процесс;
 - d) все ответы правильные;

е) правильного ответа нет.

2. Что входит в состав диаграмм:

а) блоки, описывающие подпроцессы (функции системы), и дуги, связывающие блоки вместе и изображающие взаимодействия и взаимосвязи между блоками;

б) блоки, описывающие подпроцессы (функции системы);

с) дуги, связывающие блоки вместе и изображающие взаимодействия и взаимосвязи между блоками;

д) нумерация диаграммы;

е) правильные ответы б), д).

3. Укажите этап, в котором происходит построение диаграммы процесса верхнего уровня:

а) рисование дуги управления;

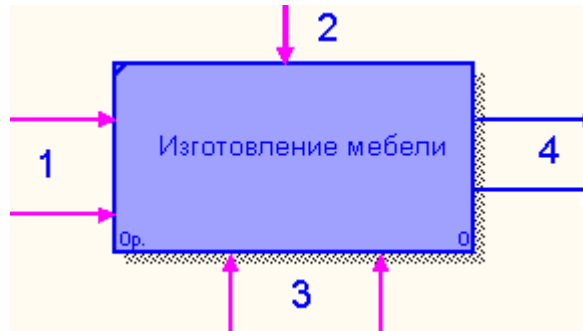
б) определение основного бизнес-процесса;

с) построение контекстной диаграммы;

д) функциональная декомпозиция каждого процесса, с помощью детализирующих диаграмм;

е) все ответы правильные.

4. Укажите, для чего предназначена левая сторона блока (1):



а) для **управления** (правила, стратегии, стандарты);

б) для **механизмов** (ресурсы, которые выполняют процесс);

с) для **выходов** (материал или информация, получаемые в результате выполнения процесса);

д) для **входов** (материал или информация, которые используются или преобразуются для получения результата);

е) правильные ответы с), д).

5. Какая команда из контекстно-зависимого меню позволит изменить стиль дуги:

а) Font Editor;

б) Color Editor;

с) Trim;

д) Style Edition;

е) Background Color;

ф) Squiggle.

6. Для чего предназначен инструмент **T**:

а) используется для создания тильды (squiggle), которая соединяет дугу с ее названием;

б) используется для выбора и определения позиции объектов, добавленных в диаграмму;

с) используется для создания текстовых блоков;

д) используется для выбора цвета текста и фона;

е) правильного ответа нет.



После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Контрольные задания.

Предложенные задания предназначены для самостоятельной разработки модели информационной системы. Номер варианта задания для каждого учащегося учитель назначает сам. В ходе выполнения задания учащиеся должны составить модель информационной системы для автоматизации рабочего места участника процесса:

- 1) описать главную цель, его функцию;
- 2) определить основной бизнес-процесс;
- 3) построить контекстную диаграмму;
- 4) построить диаграммы верхнего уровня;
- 5) произвести функциональную декомпозицию каждого процесса с помощью детализирующих диаграмм;
- 6) составить отчет;
- 7) построить DFD - диаграмму;
- 8) построить FEO – диаграмму;
- 9) построить IDEF3 – диаграмму;
- 10) произведите стоимостный анализ разработанной модели, сгенерируйте отчет.

Вариант 1.

Разработать модель информационной системы для автоматизации рабочего места библиотекаря.

Вариант 2.

Разработать модель информационной системы для автоматизации рабочего места секретаря школы.

Вариант 3.

Разработать модель информационной системы для автоматизации рабочего места регистратора в поликлинике.

Вариант 4.

Разработать модель информационной системы для автоматизации рабочего места страхового агента.

Вариант 5.

Разработать модель информационной системы для автоматизации рабочего места администратора гостиницы

После того как учитель определит ваш вариант работы, начинайте его выполнение. Это задание вы будете делать на протяжении всех лабораторных работ, постепенно изучая программную среду BPwin и отрабатывая навыки работы в ней.

Приступая к выполнению контрольного задания:

1. Сформулируйте целевые функции работника предприятия.
2. Определите подразделения предприятия, нормативные документы.
3. Определить входную информацию (данные или материальные ресурсы), которая преобразуется в процессе для получения результата.
4. Определите выходную информацию - готовый результат.
5. Укажите механизмы, которые выполняют процесс.
6. Создайте новый файл в программе **BPwin**.
7. Постройте контекстную диаграмму, произведите ее форматирование.
8. Сохраните новый файл в своей папке с именем **Kr1.bp1**.

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 8

Тема: Декомпозиция контекстной диаграммы

Цель: научиться производить декомпозицию контекстной диаграммы; освоить правила построения дуг и тоннелирования стрелок.

Теоретические сведения

Декомпозиция

 После создания контекстной диаграммы, которая представляет собой описание контекста моделируемой системы, проводится **функциональная декомпозиция** – система разбивается на подсистемы и каждая подсистема описывается в том же синтаксисе, что и система в целом. Затем каждая подсистема разбивается на более мелкие и так до достижения нужного уровня подробности. В результате такого разбиения, каждый фрагмент системы изображается на отдельной диаграмме декомпозиции. *Диаграмма декомпозиции предназначена для детализации работы.*

При декомпозиции процесса все стрелки, входящие или исходящие из него, должны быть перенесены на диаграмму нижнего уровня и использованы при ее построении. При этом запрещены всякие новые стрелки, выходящие за пределы новой диаграммы, кроме специальных, так называемых "**тоннелированных**" стрелок.

Создание диаграммы A0

Диаграмма верхнего уровня создается путем декомпозиции основной функции контекстной диаграммы. На диаграмме декомпозиции функции нумеруются автоматически слева направо. Номер функции показывается в правом нижнем углу. В левом верхнем исчезает небольшая диагональная черта, которая показывает, что данная функция была декомпозирована.

Практическое задание



Задание 10. Детализация процесса «Изготовление мебели».

Откройте файл **Lab1.bp1**, сохраненный на предыдущем уроке.

Следующим шагом является детализация контекстного процесса с помощью диаграммы верхнего уровня. Эта диаграмма содержит в себе четыре процесса:

- 1) Процесс 1.1 – ПЕРЕРАБОТКА СЫРЬЯ.
- 2) Процесс 1.2 – ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ.
- 3) Процесс 1.3 – СБОРКА ИЗДЕЛИЯ.
- 4) Процесс 1.4 – КОНТРОЛЬ КАЧЕСТВА.

Произведите детализацию процесса «Изготовление мебели», задав нужное количество новых блоков. Для этого:

1. Щелкните по блоку «Изготовление мебели» и выберите инструмент .
2. В диалоговом окне введите число, на которое будет произведена декомпозиция – 4.
3. Укажите тип диаграммы **IDEF0** (рис. 2.1.) и нажмите ОК.

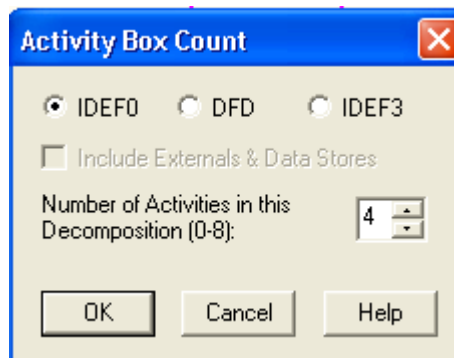


Рис. 2.1. Диалоговое окно декомпозиции блока

4. Укажите названия новых блоков («Переработка сырья», «Изготовление деталей», «Сборка изделия», «Контроль качества»).

При декомпозиции функции входящие в нее и исходящие из нее дуги автоматически появляются на диаграмме декомпозиции (миграция дуг), но при этом не касаются блоков. Такие стрелки называются *несвязанными* и воспринимаются в BRwin как синтаксическая ошибка (см. рис.2.2.).

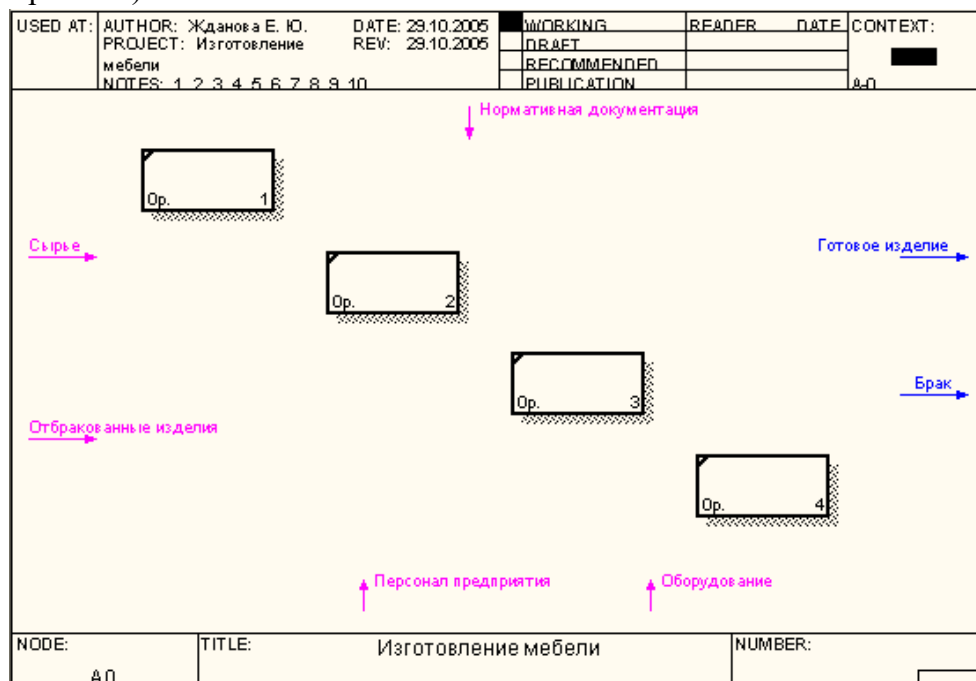


Рис. 2.2. Декомпозиция верхнего уровня

Определим входные и выходные потоки для новых процессов.

Процесс 1.1. ПЕРЕРАБОТКА СЫРЬЯ:

- 1) Вход – СЫРЬЁ.
- 2) Вход – ОТБРАКОВАННЫЕ ИЗДЕЛИЯ.
- 3) Выход – ЗАГОТОВКИ.

Произведем процесс связывания мигрирующих дуг:

5. Выберите инструмент  рисования дуг.
 6. Щелкните мышью по наконечнику входного потока СЫРЬЁ.
 7. Щелкните по входной стороне блока ПЕРЕРАБОТКА СЫРЬЯ.
- Вход – ОТБРАКОВАННЫЕ ИЗДЕЛИЯ построим немного позже.
- Для построения выходного потока ЗАГОТОВКИ выполните действия:

8. Выберите инструмент  рисования дуг.
9. Щелкните левой кнопкой мышки по выходной стороне блока ПЕРЕРАБОТКА СЫРЬЯ.
10. Затем щелкните по входной стороне блока ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ.

11. Выберите инструмент  текст, в контекстном меню – команду **Name**, укажите название дуги ЗАГОТОВКИ.
12. Проверьте себя (рис. 2.3.).

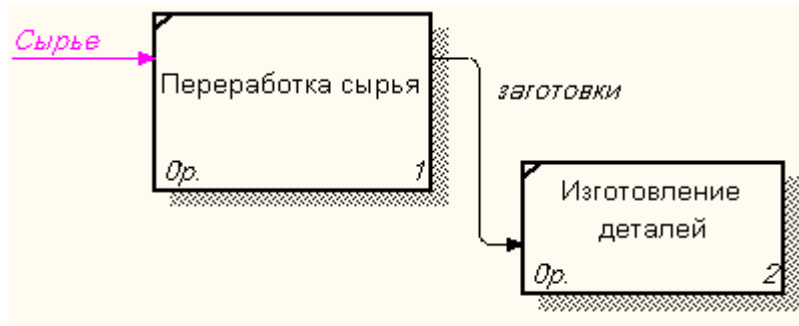


Рис. 2.3. Фрагмент диаграммы



Задание 11. Детализация процесса «Изготовление мебели».

1. Самостоятельно выполните детализацию процессов:

Процесс 1.2. ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ:

- 1) Вход – ЗАГОТОВКИ.
- 2) Выход – ГОТОВЫЕ ДЕТАЛИ.

Процесс 1.3. СБОРКА ИЗДЕЛИЯ:

- 1) Вход – ГОТОВЫЕ ДЕТАЛИ.
- 2) Выход – СОБРАННОЕ ИЗДЕЛИЕ.

Процесс 1.4. КОНТРОЛЬ КАЧЕСТВА:

- 1) Вход – СОБРАННОЕ ИЗДЕЛИЕ.
- 2) Выход – ГОТОВОЕ ИЗДЕЛИЕ.
- 3) Выход – БРАК.
- 4) Выход – ПРОИЗВОДСТВЕННЫЕ ОТХОДЫ



После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 12. Смена направления дуги.

На Выходе БРАК не выходит за границу модели, а возвращается в процесс ПЕРЕРАБОТКА СЫРЬЯ:

1. Удалите дуги ОТБРАКОВАННЫЕ ИЗДЕЛИЯ и БРАК.
2. Выберите инструмент  рисование дуг.
3. Щелкните левой кнопкой мыши на Выходе блока КОНТРОЛЬ КАЧЕСТВА.
4. Щелкните левой кнопкой на Входе блока ПЕРЕРАБОТКА СЫРЬЯ.
5. Назовите новую дугу – БРАК (рис. 2.4.).

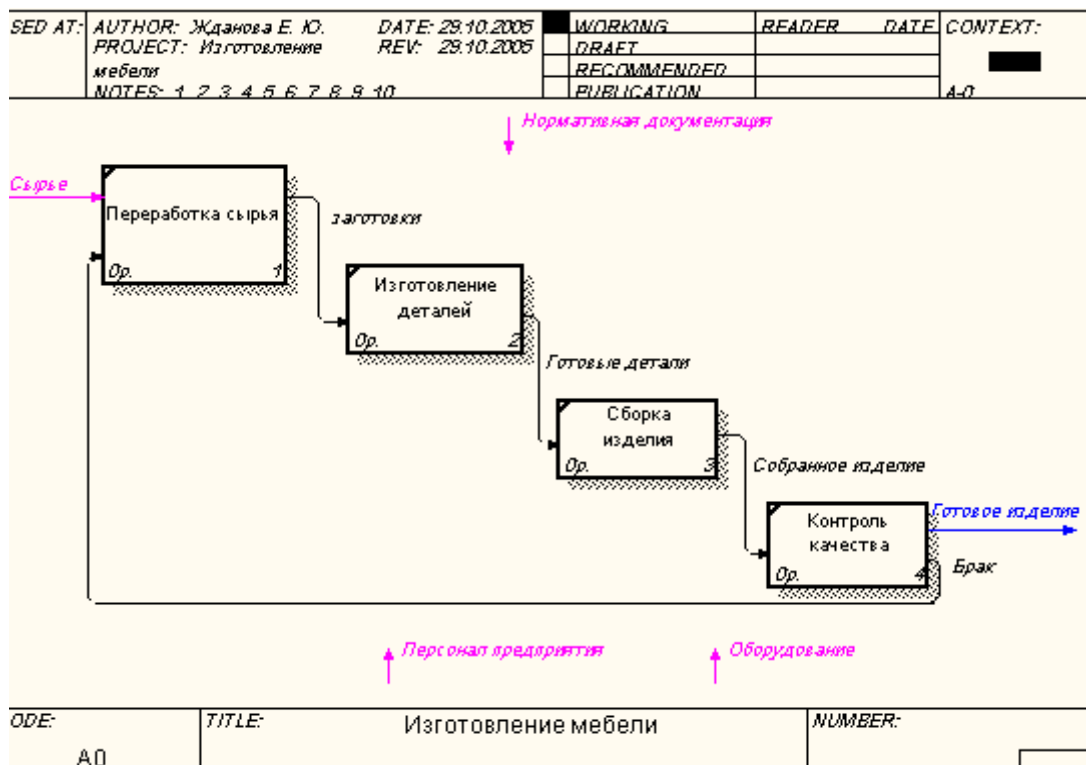


Рис. 2.4. Процесс декомпозиции



Задание 13. Построение ответвлений дуг.

Переработка сырья, изготовление деталей, сборка изделия, контроль качества осуществляются согласно Нормативным документам, поэтому у управляющей стрелки НОРМАТИВНАЯ ДОКУМЕНТАЦИЯ появятся ответвления: НОРМЫ ПЕРЕРАБОТКИ СЫРЬЯ, ЧЕРТЕЖ ДЕТАЛИ, СБОРОЧНЫЙ ЧЕРТЕЖ, СТАНДАРТ КАЧЕСТВА.

1. Выберите инструмент рисование дуг.
2. Щелкните мышью по наконечнику входного потока НОРМАТИВНАЯ ДОКУМЕНТАЦИЯ.
3. Щелкните по входной стороне блока ПЕРЕРАБОТКА СЫРЬЯ.
4. Самостоятельно выполните ответвления дуги НОРМАТИВНАЯ ДОКУМЕНТАЦИЯ на блоки ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ, СБОРКА ИЗДЕЛИЯ, КОНТРОЛЬ КАЧЕСТВА.
5. Проверьте себя (рис. 2.5.).

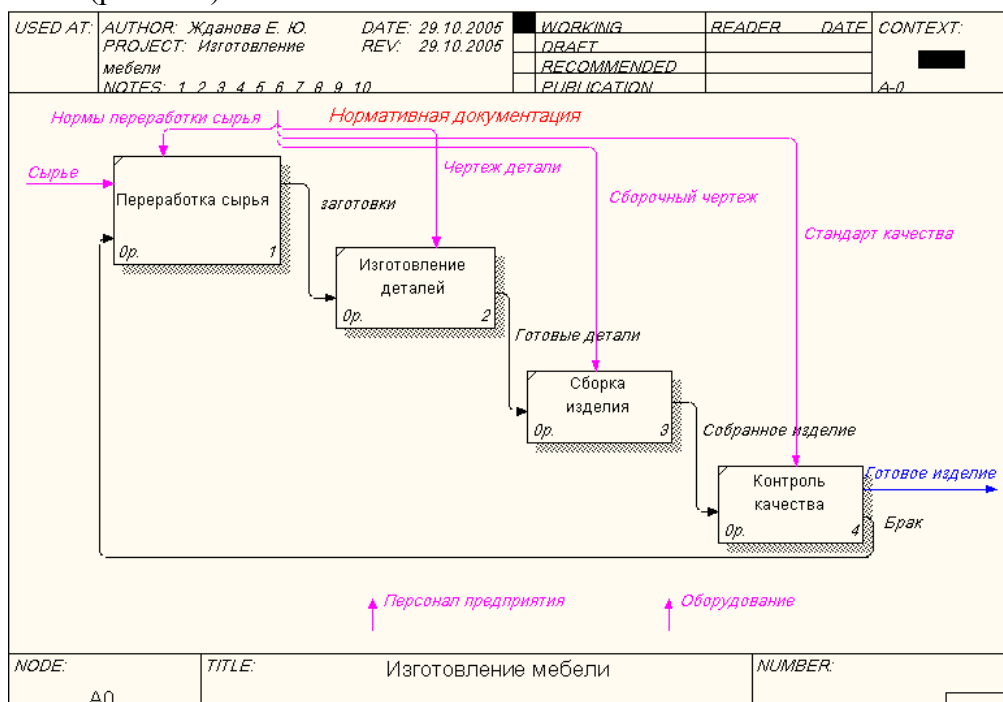


Рис. 2.5. Ответвления дуги НОРМАТИВНАЯ ДОКУМЕНТАЦИЯ



Задание 14. Построение дуг Персонал предприятия, Оборудование.

Дуги ПЕРСОНАЛ ПРЕДПРИЯТИЯ и ОБОРУДОВАНИЕ для всех процессов будут одинаковыми.

1. Самостоятельно соедините каждую дугу с каждым блоком, укажите ее имя.
2. Проверьте себя (рис. 2.6.).

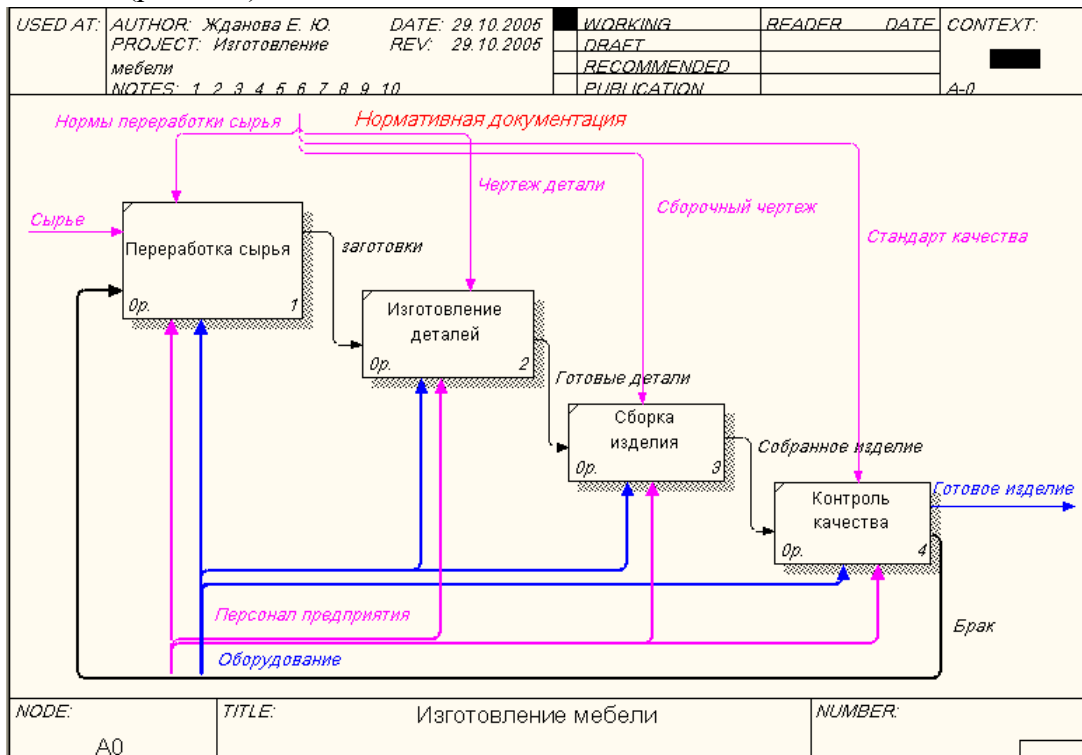


Рис. 2.6. Построение дуг ПЕРСОНАЛ ПРЕДПРИЯТИЯ и ОБОРУДОВАНИЕ



После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 15. «Тоннелирование» стрелок.

1. В Процесе 1.2. ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ постройте новую граничную дугу, которой обозначьте Выход – ПРОИЗВОДСТВЕННЫЕ ОТХОДЫ.

Вновь внесенные граничные дуги на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках и автоматически не появляются на диаграмме верхнего уровня.

Для их «перетаскивания» вверх нужно:

2. Выбрать инструмент редактирования.
3. Кликнуть правой кнопкой мыши по квадратным скобкам.
4. Выбрать в контекстном меню пункт **Arrow Tunnel**.
5. В появившемся диалоге **Border Arrow Editor** (рис. 2.7.) щелкнуть по кнопке **Resolve it to border arrow** для миграции стрелки на диаграмму верхнего уровня или по кнопке **Change it to resolved rounded tunnel** для «тоннелирования» дуги.

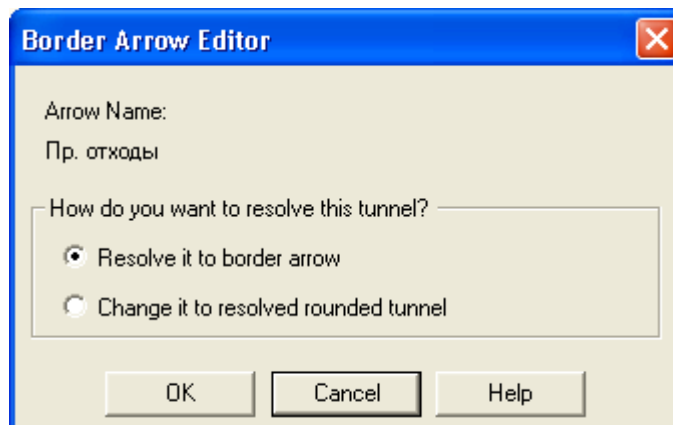


Рис. 2.7. Диалог Border Arrow Editor

Тоннельная дуга изображается с круглыми скобками на конце и не попадет на другую диаграмму (рис. 2.8.). Такое тоннелирование может быть применено для изображения малозначимых стрелок.

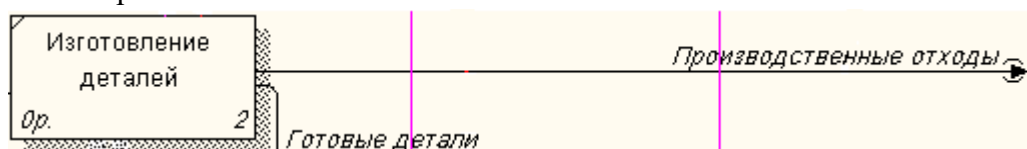


Рис. 2.8. Граничная дуга

6. Отправьте созданную дугу "Производственные отходы" в тоннель.



Задание 16. Создание обратной связи по управлению.

Качество изделия может быть повышено путем непосредственного регулирования процессами изготовления деталей и сборки мебели в зависимости от результата (выхода) работы КОНТРОЛЬ КАЧЕСТВА.

Обратная связь по управлению свидетельствует об эффективности бизнес-процесса и создается следующим образом:

1. Выберите инструмент  рисование дуг.
2. Щелкните мышью по выходу КОНТРОЛЬ КАЧЕСТВА.
3. Щелкните по управлению блоков ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ, СБОРКА ИЗДЕЛИЯ.
4. Выберите инструмент  текст.
5. Назовите обратную связь РЕКОМЕНДАЦИИ.

После выполнения работы у вас должна получиться следующая диаграмма (рис. 2.9.):

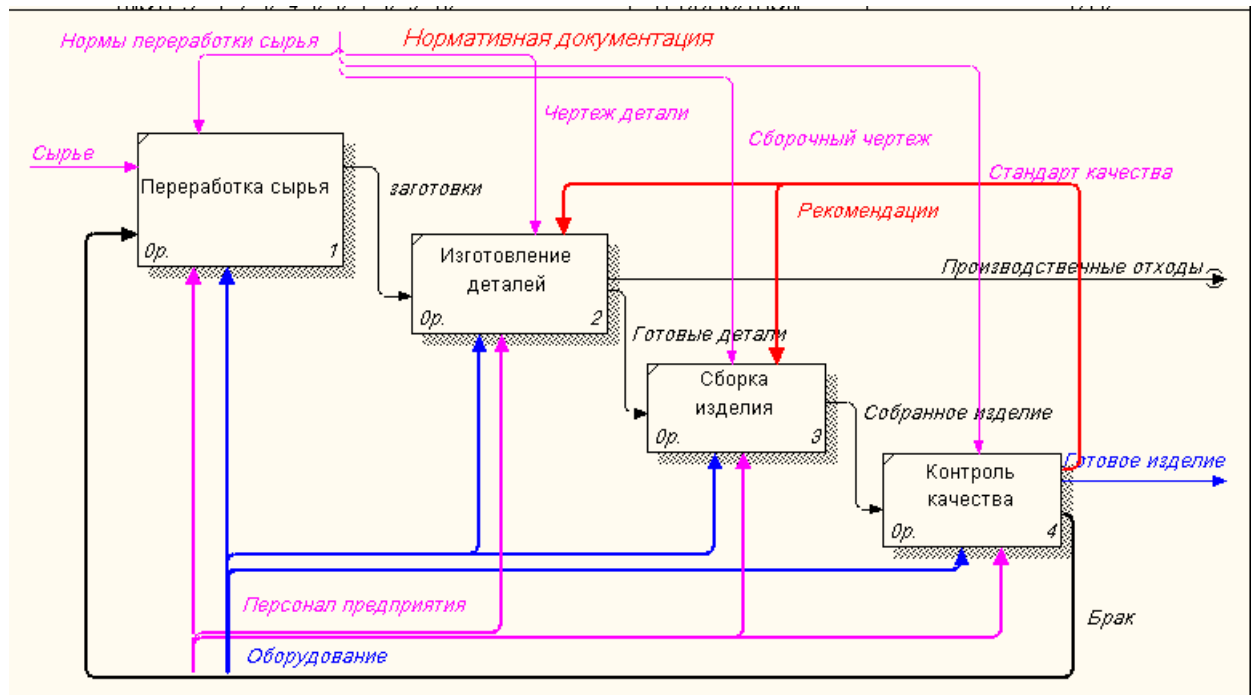


Рис. 2.9. Диаграмма декомпозиции блока ИЗГОТОВЛЕНИЕ МЕБЕЛИ



Задание 17. Сохранение полученной диаграммы.

1. В меню **File** выберите **Save as**.
2. Укажите путь к своей папке и имя файла **Lab2.bp1**.
3. Нажмите **OK**.



После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Контрольные вопросы

1. Как создается диаграмма верхнего уровня?
2. Как на диаграмме отображается декомпозиция?
3. Каким инструментом задается количество блоков для декомпозиции?
4. Какие стрелки VPwin воспринимает как синтаксическую ошибку?
5. Для чего создается обратная связь по управлению?
6. Для чего служит тоннелирование дуг?



Тестовое задание

1. В каком порядке надо располагать блоки на диаграмме детализации:
 - a) в произвольном;
 - b) обязательно справа налево;
 - c) в соответствии с уровнем сложности последующей детализации;
 - d) в соответствии с их доминированием;
 - e) все ответы правильные.
2. Для чего предназначена диаграмма детализации процесса:
 - a) для обозначения тоннелированных дуг;
 - b) для описания контекста моделируемой системы;
 - c) для детализации работы;
 - d) все ответы правильные.
3. Как создается диаграмма верхнего уровня:

- a) путем декомпозиции одной из функций контекстной диаграммы;
- b) путем декомпозиции основной функции контекстной диаграммы;
- c) путем декомпозиции диаграмм нижнего уровня;
- d) путем декомпозиции одной из функций на диаграмме верхнего уровня;
- e) правильного ответа нет.

4. Какой инструмент необходимо выбрать, чтобы с его помощью произвести декомпозицию блока?

- a)  ;
- b)  ;
- c)  ;
- d)  ;
- e)  .

5. Каким инструментом производится процесс связывания мигрирующих дуг?

- a)  ;
- b)  ;
- c)  ;
- d)  ;
- e)  .

6. Мигрирующая дуга – это дуга, которая...

- a) создает обратную связь по управлению;
- b) переходит из родительской диаграммы и не касается блока;
- c) обязательно тоннелируется;
- d) создает обратную связь по входу;
- e) правильного ответа нет.

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Контрольное задание

Продолжите выполнение контрольного задания.

1. Откройте файл **Kr1.bp1**.
2. Произведите детализацию бизнес-процесса.
3. Сохраните файл в своей папке с именем **Kr2.bp1**.

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 9

Тема: Построение функциональной модели. Описание модели

Цель: научиться детализировать процессы; освоить правила описания свойств модели; научиться составлять отчет о свойствах модели.

Теоретические сведения

Последним шагом построения модели является **функциональная декомпозиция**. Построенная диаграмма верхнего уровня также имеет множество процессов, которые в свою очередь могут быть детализированы в диаграммы нижнего уровня. Таким образом строится иерархия IDEF0 с контекстной диаграммой в вершине иерархии.

Этот процесс декомпозиции продолжается до достижения нужного уровня подробности. При таком построении иерархии IDEF0 каждый процесс более низкого уровня необходимо соотнести с процессом верхнего уровня. Обычно для этой цели все работы модели нумеруются. Номер состоит из префикса и числа. Может быть использован префикс любой длины, но обычно используют префикс А.

Контекстная работа дерева имеет номер А0. Работы декомпозиции А0 имеют номера А1, А2, А3 и т.д. Работы декомпозиции нижнего уровня имеют номер родительской работы и очередной порядковый номер, например работы декомпозиции А3 будут иметь номера А31, А32, А33, А34 и т. д.

Работы образуют иерархию, где каждая работа может иметь одну родительскую и несколько дочерних работ, образуя дерево. Такое дерево называют **деревом узлов**, а вышеописанную нумерацию - **нумерацией по узлам**.

Имеются незначительные варианты нумерации, которые можно настроить во вкладке **Numbering** (рис. 3.1.) диалога **Model Properties** (меню **Model – Model Properties**).

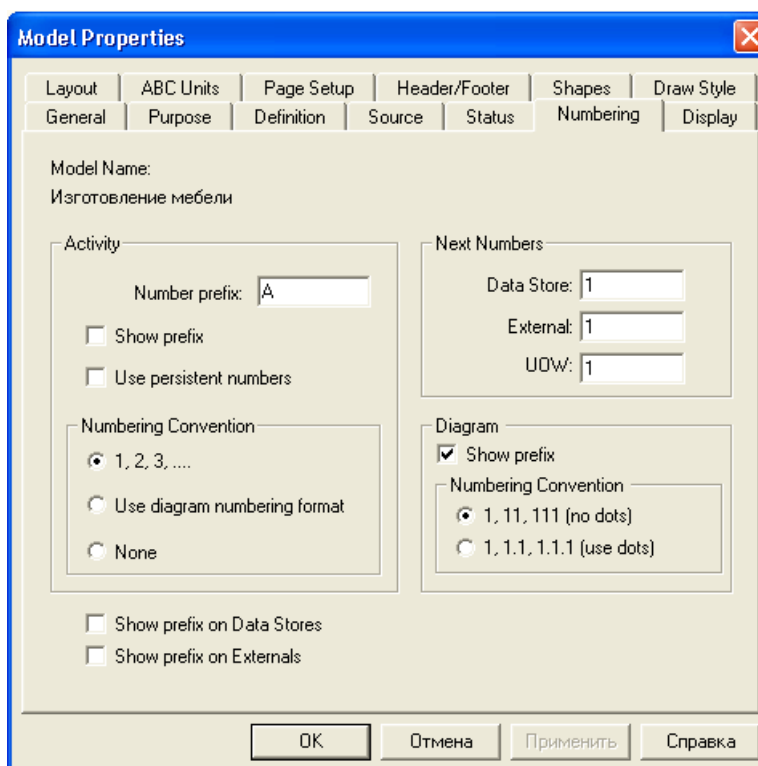


Рис. 3.1. Диалоговое окно настройки нумерации работ в диаграмме

Диаграммы **IDEF0** имеют двойную нумерацию. Во-первых, диаграммы имеют номера по узлу. Контекстная диаграмма всегда имеет номер А-0, декомпозиция контекстной

диаграммы - номер A0, остальные диаграммы декомпозиции - номера по соответствующему узлу (например, A1, A2, A21, A213 и т.д.).

ВРwin автоматически поддерживает нумерацию по узлам, т. е. при проведении декомпозиции создается новая диаграмма и ей автоматически присваивается соответствующий номер. В результате проведения экспертизы диаграммы могут уточняться и изменяться, следовательно, могут быть созданы различные версии одной и той же (с точки зрения ее расположения в дереве узлов) диаграммы декомпозиции. ВРwin позволяет иметь в модели только одну диаграмму декомпозиции в данном узле. Прежние версии диаграммы можно хранить в виде бумажной копии либо как FEO-диаграмму. (К сожалению, при создании FEO-диаграмм отсутствует возможность отката, т. е. можно получить из диаграммы декомпозиции FEO, но не наоборот.)

В любом случае следует отличать различные версии одной и той же диаграммы. Для этого существует специальный номер - C-number, который должен присваиваться автором модели вручную. C-number - это произвольная строка, но рекомендуется придерживаться стандарта, когда номер состоит из буквенного префикса и порядкового номера, причем в качестве префикса используются инициалы автора диаграммы, а порядковый номер отслеживается автором вручную, например **ЖЕЮ00021** (рис. 3.2.).

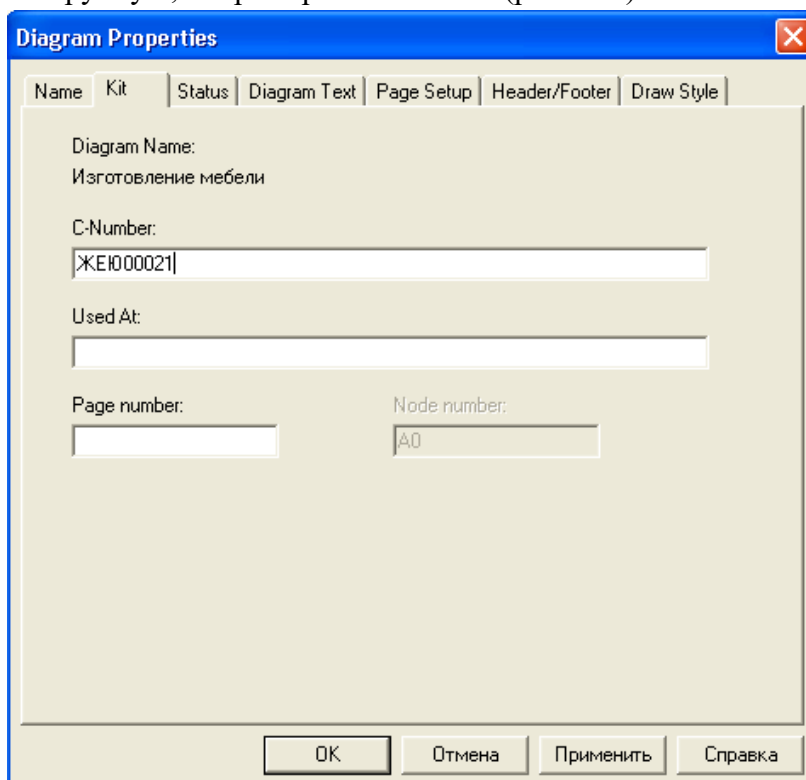


Рис. 3.2. Диалоговое окно присваивания номера данной версии диаграммы

Практическое задание

На предыдущих лабораторных работах вы построили контекстную диаграмму процесса "Изготовление мебели" и провели его детализацию с помощью диаграммы верхнего уровня. Последним шагом построения модели является **функциональная декомпозиция**, т.е. разбиение сложных процессов на более простые. Этот процесс декомпозиции продолжается до достижения нужного уровня подробности.



Задание 18. Детализация процесса «Изготовление деталей».

1. Откройте файл **Lab2.bp1**, сохраненный на предыдущем уроке.
2. Проведите детализацию процесса **1.2. ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ** с помощью диаграммы нижнего уровня. Данные представлены в таблице 3.1:

Таблица 3.1. Детализирование процесса «Изготовление деталей»

Процесс	Вход	Выход
1.2.1 – Переработка заготовки в деталь	Заготовки	Готовые детали
1.2.2 – Проверка качества деталей	Готовые детали	Готовые детали, брак
Управляющие стрелки и стрелки механизмов, указанные на диаграмме верхнего уровня должны быть и в диаграмме детализации.		

- Выберите инструмент  и щелкните по блоку ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ;
- В диалоговом окне введите число, на которое будет произведена декомпозиция - 2;
- Укажите тип диаграммы **IDEF0** (рис. 3.3.) и нажмите ОК.

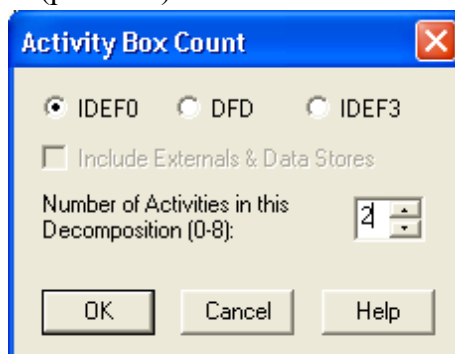


Рис. 3.3. Диалоговое окно декомпозиции блока

Вы получите диаграмму декомпозиции уровня A2 (рис. 3.4.).

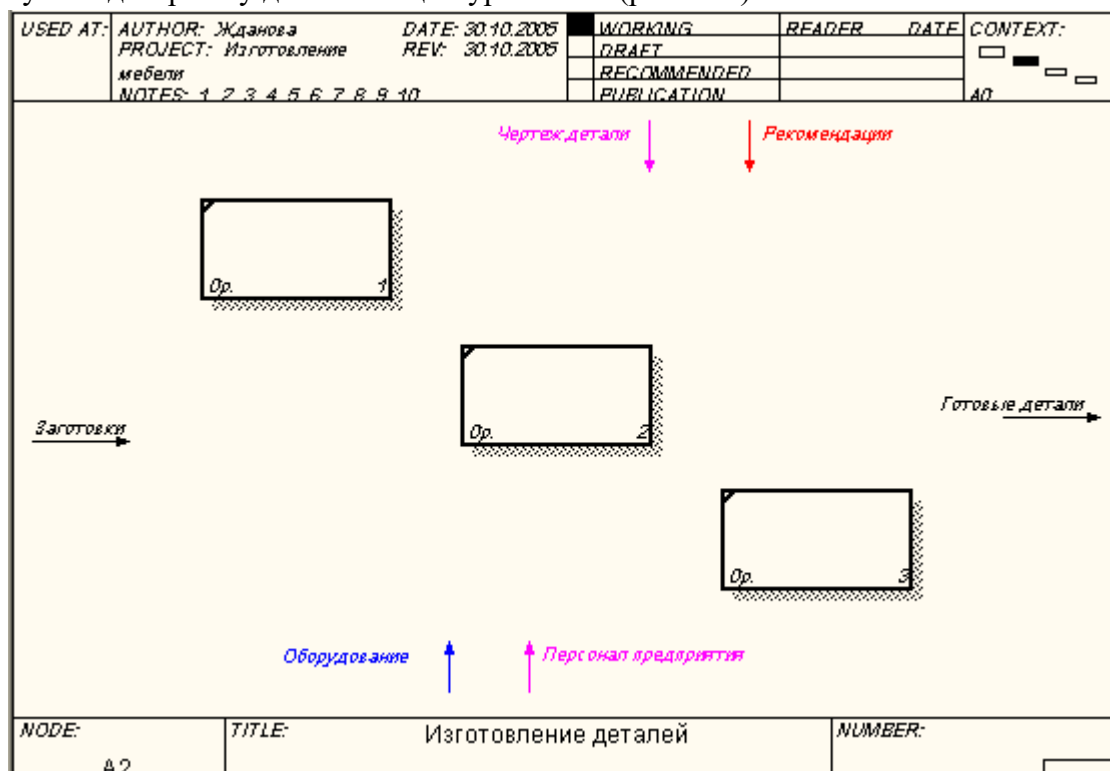
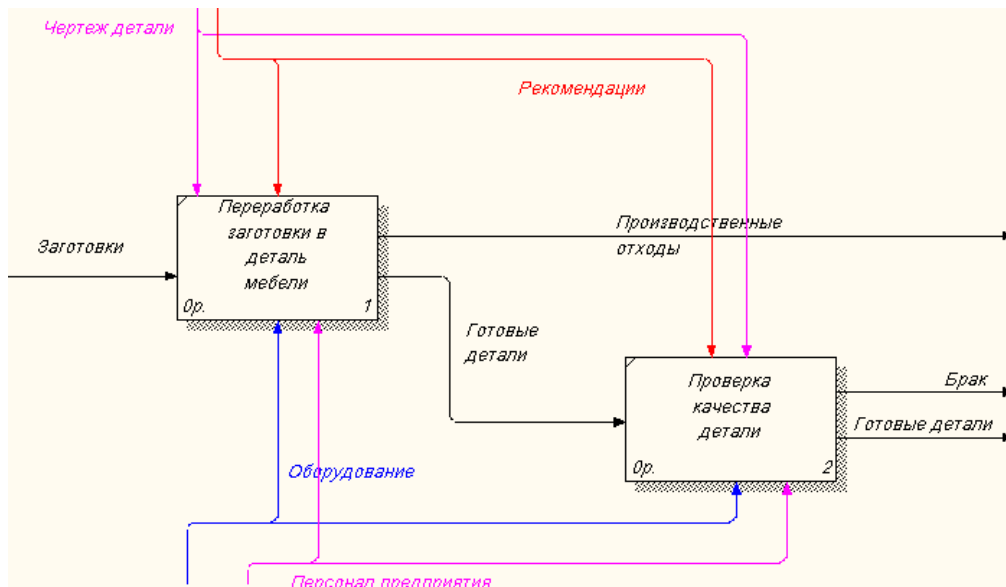
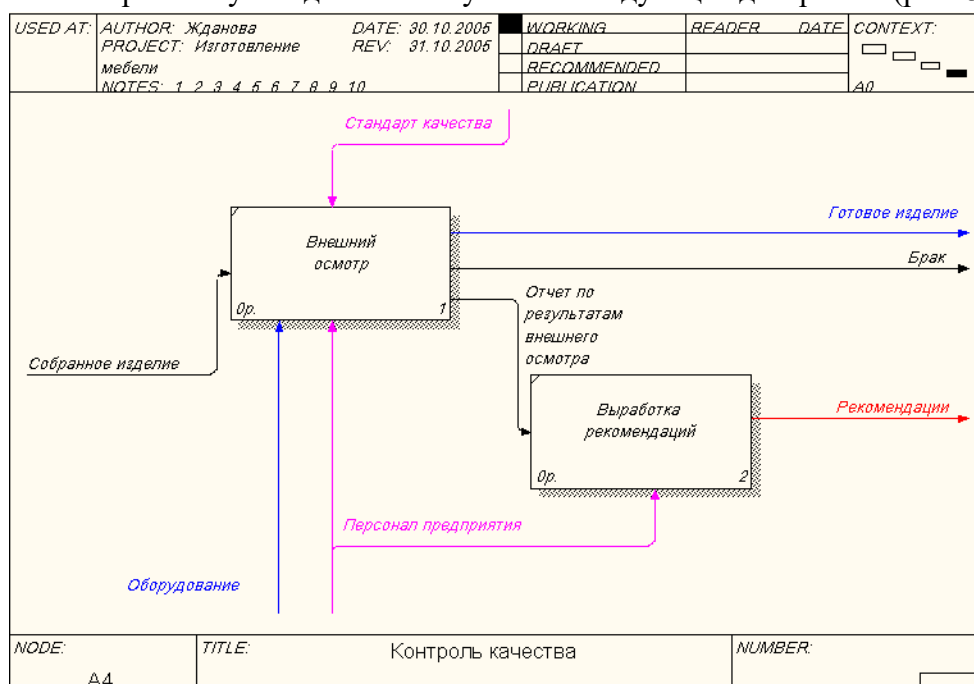


Рис. 3.4. Декомпозиция уровня A2

- Укажите названия процессов;
- Соедините дугами обозначенные процессы, используя данные из таблицы 3.1;
- Проверьте себя (рис. 3.5.).



1. Самостоятельно выполните детализацию процесса КОНТРОЛЬ КАЧЕСТВА.
После выполнения работы у вас должна получиться следующая диаграмма (рис. 3.6.):



 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



IDEF0-модель предполагает наличие четко сформулированной цели, единственного субъекта моделирования и одной точки зрения. Для внесения области, цели и точки зрения в модели IDEF0 в BPwin следует:

1. Выбрать пункт меню **Model - Model Properties**, вызывающий диалог **Model Properties** (рис. 3.7.);

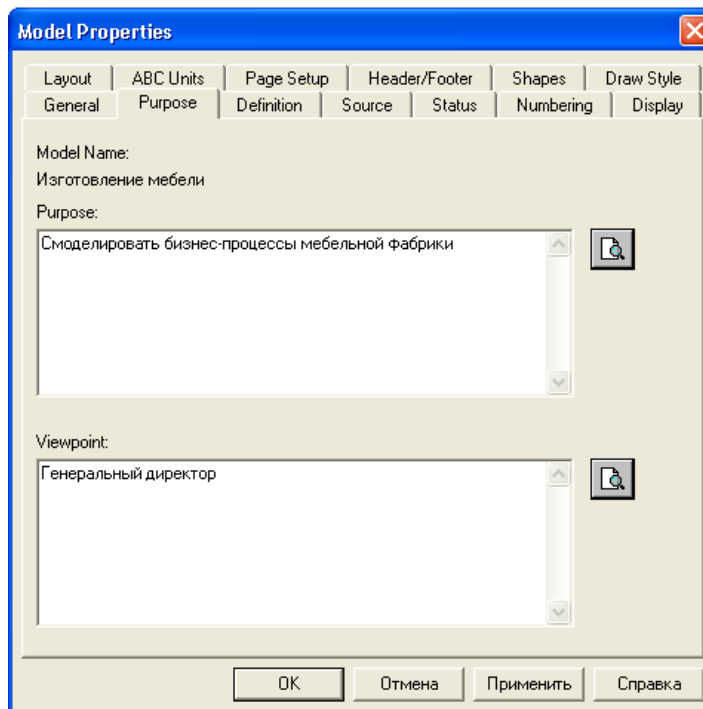


Рис. 3.7. Диалог задания свойств модели

- Во вкладку **Purpose** внести цель и точку зрения, а во вкладку **Definition** – определение модели;

Цель и точку зрения принято выносить на контекстную диаграмму А-0 в виде текстового блока. После описания они появятся на контекстной диаграмме в виде текстового блока. Описание производится на уровне контекстной диаграммы.

Для описания цели и точки зрения следует:

- Перейти на уровень диаграммы А-0;
- Выбрать кнопку текста **T** на палитре инструментов;
- Щелкнуть мышью в позиции предполагаемого ввода текста;
- В диалоговом окне набрать нужный текст и установить опцию значимости (обычный текст, цель или точка зрения) (рис. 3.8.).

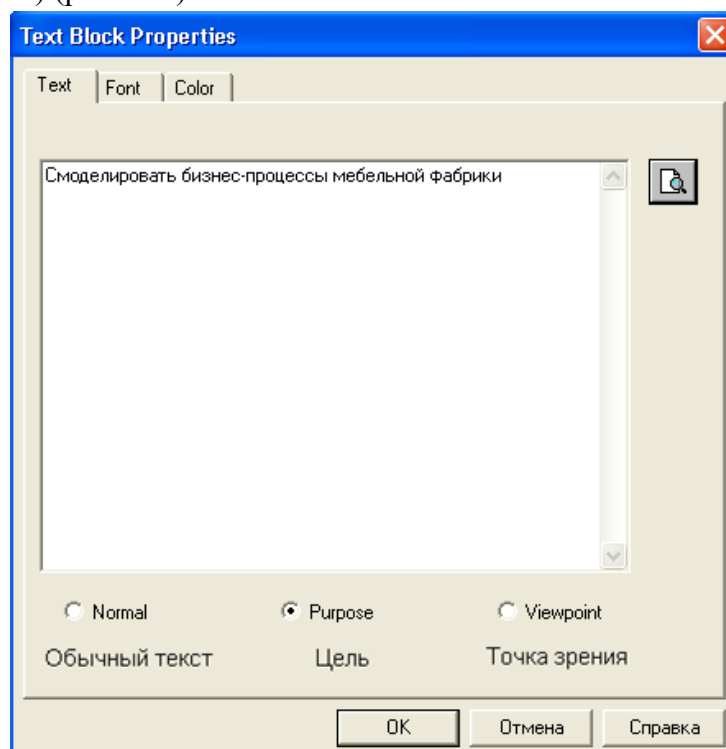


Рис. 3.8. Установка опции Text

7. Во вкладке **Status** того же диалога опишите статус модели (черновой вариант, рабочий, окончательный и т.д.), время создания и последнего редактирования (отслеживается в дальнейшем автоматически по системной дате);
8. Во вкладке **Source** опишите источники информации для построения модели (например, «Опрос экспертов предметной области и анализ документации»);
9. Вкладка **General** служит для внесения имени проекта и модели, имени и инициалов автора и временных рамок модели.



Задание 21. Составление отчета.

Результат описания модели можно получить в отчете **Model Report**.

1. Диалоговое окно настройки отчета по модели вызовите из пункта меню **Tools – Reports – Model Report**.
2. Выберите необходимые поля, при этом автоматически отображается очередность вывода информации в отчете (рис. 3.9.);

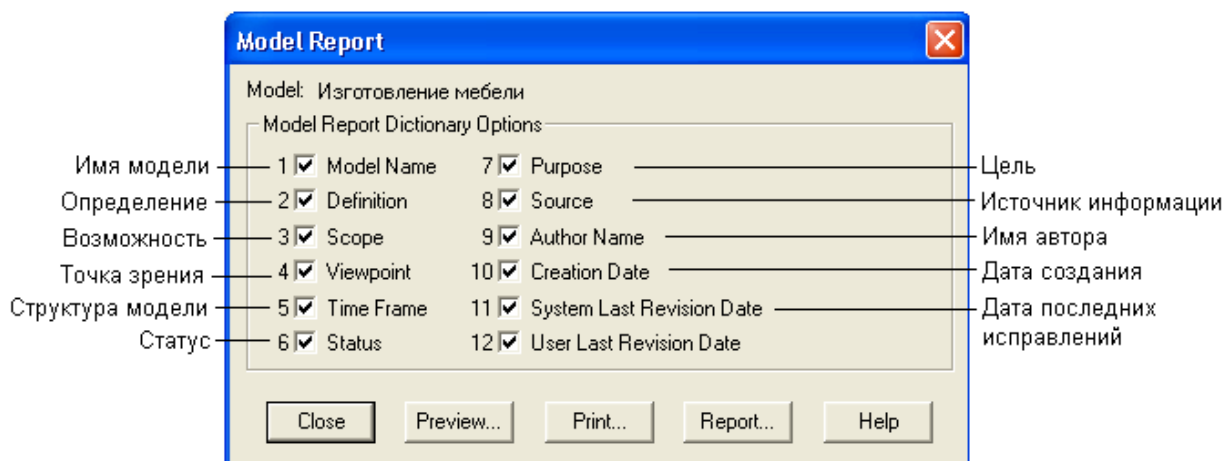


Рис. 3.9. Диалоговое окно выбора информации для отчета

3. Нажмите на кнопку **Preview**, чтобы просмотреть отчет (рис. 3.10.).

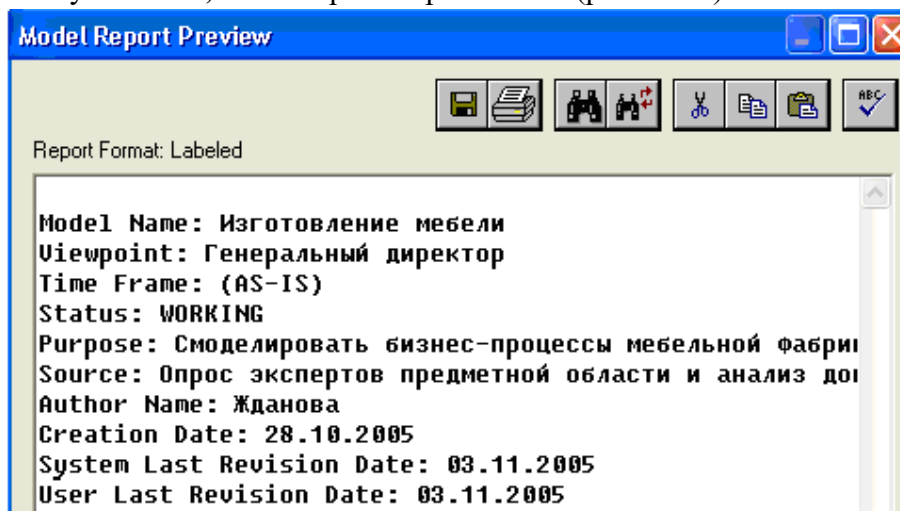


Рис. 3.10. Отчет о модели



Задание 22. Сохранение полученной диаграммы.

1. В меню **File** выберите **Save As**.
2. Укажите путь к своей папке и имя файла **Lab3.bp1**.
3. Нажмите **OK**.

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

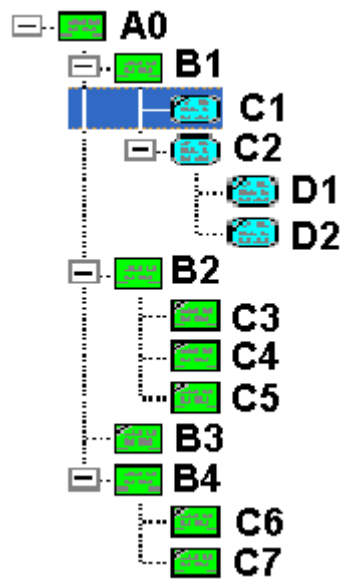
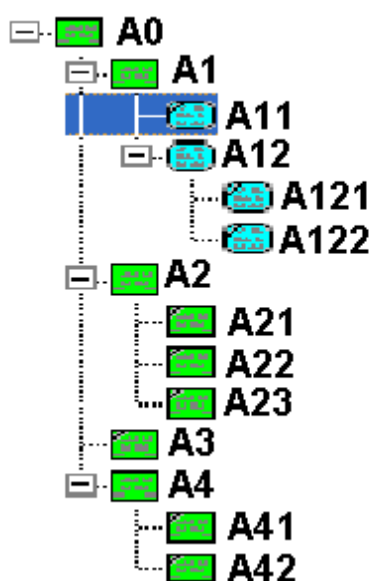
Контрольные вопросы

1. Как нумеруются модели в иерархии **IDEF0**?
2. Дайте понятие определению **Дерево узлов**.
3. Какой процесс в разработке модели называют функциональной декомпозицией?
4. Как можно вынести цель и точку зрения проекта на диаграмму?
5. Для чего необходимо составление отчета?



Тестовое задание

1. **IDFE0** - это:
 - a) диаграмма потока данных;
 - b) диаграмма бизнес - процесса;
 - c) диаграмма сущность - связь;
 - d) диаграмма ключей;
 - e) все ответы правильные.
2. Укажите номер рисунка, который отражает правильную нумерацию по узлам:



3. Сколько контекстных диаграмм может быть в модели:

- a) одна;
- b) две;
- c) столько, сколько уровней детализации;
- d) неограниченное количество;
- e) все ответы правильные.

4. Дуги управления указывают на:

- a) готовый результат;
- b) промежуточный результат;
- c) ресурсы, которые выполняют процесс;
- d) правила, стандарты;
- e) информацию для получения результата;

f) все ответы правильные.

5. Какой инструмент позволяет внести цель и точку зрения на контекстную диаграмму?

- a)  ;
- b)  ;
- c)  ;
- d)  ;
- e)  .

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Контрольное задание

Продолжите выполнение контрольного задания.

1. Откройте файл **Kr2.bp1**.
2. Произведите декомпозицию 2-х бизнес-процессов.
3. Создайте описание модели.
4. Сохраните файл в своей папке с именем **Kr3.bp1**.

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №10

Тема: Знакомство с Rational Rose

Цель: ознакомиться с приложением Rational Rose

Rational Rose — это CASE -система для визуального моделирования объектно-ориентированных программных продуктов.

Визуальное моделирование — процесс графического описания разрабатываемого программного обеспечения.

Ознакомьтесь с окном Rational Rose (рис. 1):

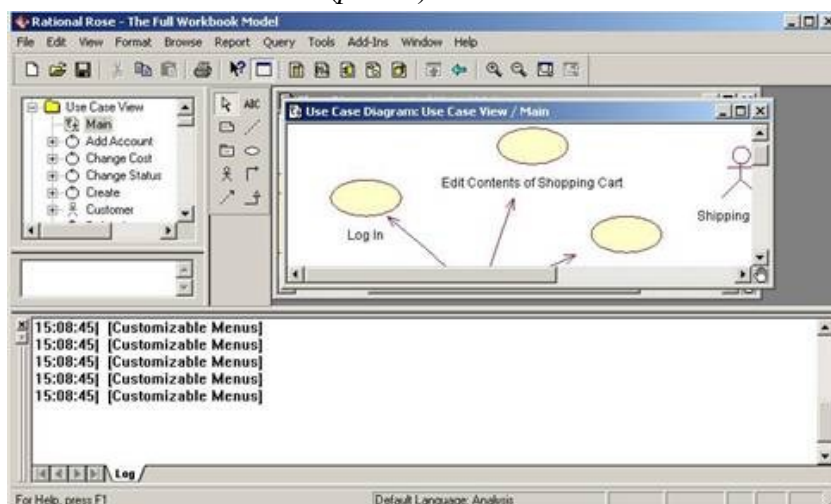


Рис. 1. Окно Rational Rose

- *браузер программы* - располагается под строкой меню и строкой инструментов в левой части окна приложения. Браузер (навигатор) является инструментом навигации по иерархической структуре модели. Знак + рядом с папкой означает, что в ней имеются дополнительные элементы;

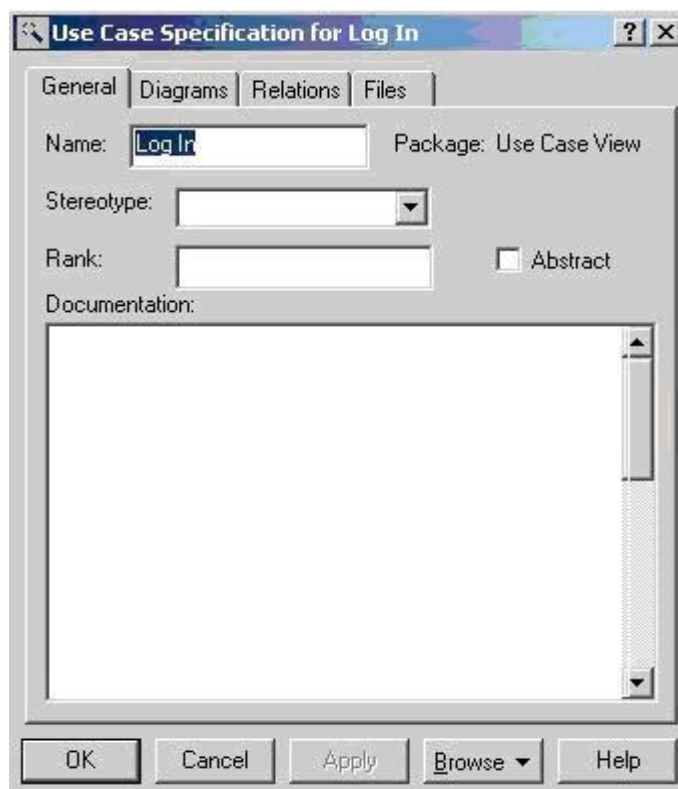


Рис. 2. Окно спецификаций

- *панель "инструменты диаграммы"* - с права от окна браузера. Она меняется в зависимости от выбранной диаграммы;
- *окно диаграммы*, используется для создания, отображения и изменения диаграмм на языке UML. В нем находятся разные диаграммы. Активная диаграмма имеет синий заголовок;
- *окно спецификации* – позволяет задавать характеристики элемента диаграммы (рис.2);
- *окно документации* – окно для вывода словесного описания данного элемента. Описание можно вводить и в поле Documentation окна спецификации.

Как показано на *рис. 3*, кнопки строки инструментов позволяют выполнять стандартные и специальные действия.

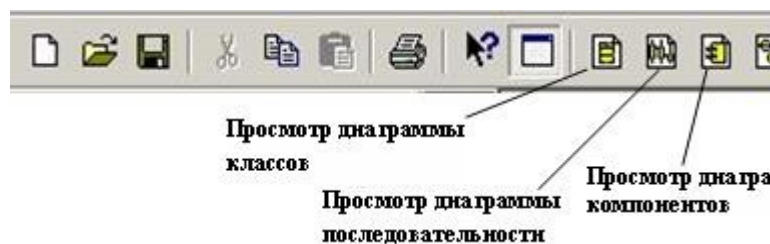


Рис. 3. Строка инструментов.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №11

Тема: Работа с Rational Rose

Изучение программного пакета Rational Rose 2000. Диаграмма прецедентов

Цель: изучение объектно-ориентированного моделирования и исследование процесса построения диаграммы прецедентов в заданной предметной области с помощью пакета Rational Rose 2000.

Порядок выполнения

Выберите тему для моделирования.

Исследуйте предметную область: определите исполнителей, выполняемые функции.

Постройте диаграмму прецедентов или, если требуется, несколько диаграмм.

Ответьте на контрольные вопросы.

Содержание отчета

1. Название работы.
2. Цель работы.
3. Анализ предметной области.
4. Диаграмма прецедентов с комментариями.
5. Спецификация прецедентов.

Диаграммы прецедентов

В соответствии с методологией объектно-ориентированного анализа и проектирования первым этапом является анализ требований, который подразумевает выделение процессов и требований и их формулировку в виде прецедентов.

Прецедент в объектном моделировании (англ. – use case) представляет собой документ, описывающий последовательность событий, связанных с исполнителем (внешним агентом), который для завершения требуемого процесса использует создаваемую систему. Прецеденты являются описанием или вариантами использования системы. С помощью прецедента описывается некоторый процесс.

По результатам анализа прецедентов на первом этапе моделирования предметной области создается диаграмма определения требований к системе Use Case (сценарии поведения). Данная диаграмма позволяет создавать диаграммы поведения объектов системы.

На диаграмме прецедентов иллюстрируется набор прецедентов системы и исполнители, а также взаимосвязи между ними. Прецеденты определяют, как исполнители взаимодействуют с программной системой. В процессе этого взаимодействия исполнителем генерируются события, передаваемые системе, которые представляют собой запросы на выполнение некоторой операции.

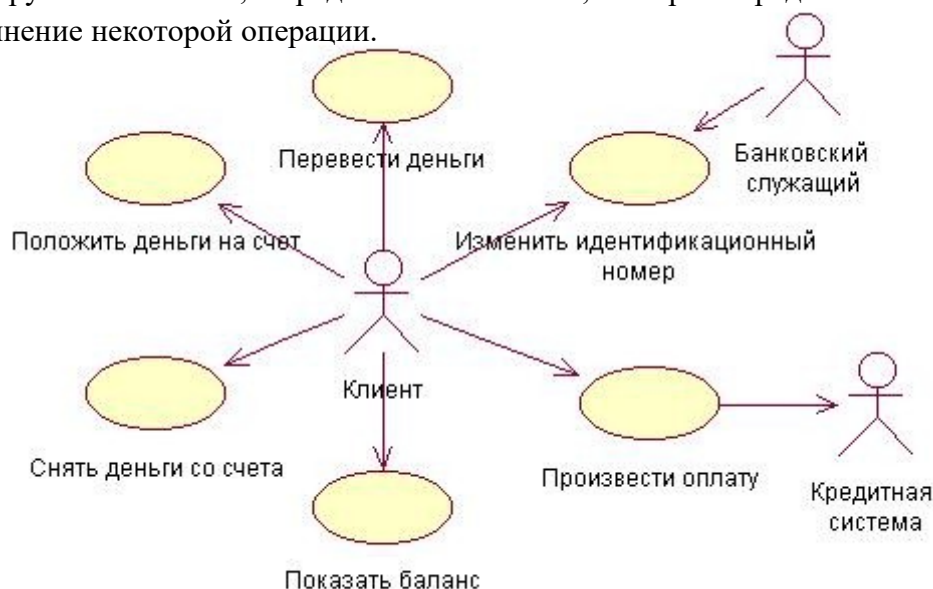


Рис.1 – Диаграмма прецедентов, описывающая процесс обслуживания клиента в банке
Диаграмма прецедентов содержит: варианты использования (прецеденты) системы (use case); действующее лицо (actor).

Диаграмма отражает взаимодействие вариантов использования и действующих лиц. Она отражает требования к системе с точки зрения пользователя.

Варианты использования системы – описание функций системы на «высоком уровне». Они описывают все, что происходит внутри области действия системы. Варианты использования иллюстрируют, как можно использовать систему. Они заостряют внимание на том, что пользователи хотят получить от системы. Каждый вариант использования представляет собой завершённую транзакцию между пользователем и системой.

Действующее лицо – все, что взаимодействует с системой, передает или получает информацию от системы. Исполнитель (actor) является внешним по отношению к системе понятием, которое определенным образом участвует в процессе, описываемом прецедентом. Они описывают все, что находится вне системы. Это пользователи системы, другие системы, взаимодействующие с описываемой, время. Каждый прецедент должен быть инициирован действующим лицом.

Как правило, отдельные шаги или виды деятельности в виде прецедента не представляются.

Часто для одной системы создается несколько диаграмм Вариантов Использования. На диаграмме высокого уровня (Main) указываются только пакеты вариантов использования. Другие диаграммы описывают совокупности вариантов использования и действующих лиц.

Цель диаграмм – документирование вариантов использования, действующих лиц и связей между ними.

Разрабатывая диаграммы, придерживаются правил:

Не моделируют связи между действующими лицами. По определению они находятся вне сферы действия системы. Связи между ними не относятся к ее компетенции.

Не соединяют стрелкой непосредственно два варианта использования (кроме связей использования и расширения). Диаграмма описывает только, какие варианты использования доступны системе, а не порядок их выполнения.

Каждый вариант использования должен быть инициирован действующим лицом. Всегда должна быть стрелка, начинающаяся на действующем лице и заканчивающаяся на варианте использования (кроме связей использования и расширения).

Думают о БД, как о слое, находящемся под диаграммой. С помощью одного варианта использования можно вводить данные в базу, а получать их – с помощью другого. Не рисуют стрелки от одного варианта к другому для изображения потока информации.

При создании диаграмм прецедентов вначале определяются исполнители (роли, пользователи).

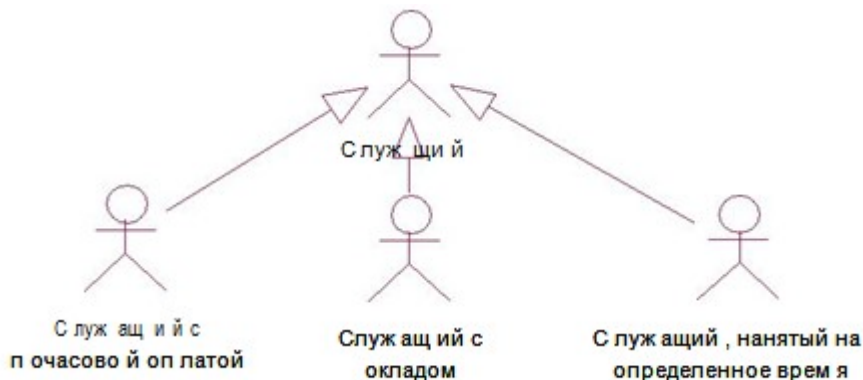


Рис.2

Исполнитель может быть абстрактным, не имеющим экземпляров. Например, есть несколько действующих лиц: служащий с почасовой оплатой, служащий с окладом и т.д. Все они являются разновидностями действующего лица - служащего. Абстрактный исполнитель существует для того, чтобы показать общность между этими типами.

Следующий шаг - идентификация прецедентов.

У каждого прецедента должно быть уникальное имя.

Каждый прецедент должен иметь связанное с ним короткое описание того, что он будет делать. Следует делать описание коротким и к «месту», при этом оно должно определять типы пользователей, выполняющих вариант использования, и ожидаемый ими конечный результат.

Предусловия варианта использования – это такие условия, которые должны быть выполнены, прежде чем вариант начнет свою работу. Например: это может быть выполнение другого варианта использования или наличие у пользователя прав доступа, требуемых для запуска данного варианта использования.

Постусловия – такие условия, которые должны быть выполнены после завершения варианта использования. С помощью постусловий можно вводить сведения о порядке выполнения вариантов использования системы. Если, скажем, после одного варианта использования должен всегда выполняться другой, это можно описать как постусловие. Такие условия имеются не у каждого варианта использования.

Конкретные детали вариантов использования отражаются в основном и альтернативном потоках событий. Поток событий поэтапно описывает, что должно происходить во время выполнения заложенной в варианты использования функциональности. Поток событий уделяет внимание тому, что (а не как) будет делать система, причем описывает это с точки зрения пользователя. Первичный и альтернативный потоки событий содержат: описание того, каким образом запускается вариант использования, различные пути выполнения варианта использования, нормальный, или основной, поток событий варианта использования, отклонения от основного потока событий (так называемые альтернативные потоки), потоки ошибок, описание того, каким образом завершается вариант использования.

Поток событий должен быть согласован с определенными ранее требованиями. Избегают детальных обсуждений того, как поток будет реализован.

Можно создать подробную спецификацию для каждого варианта использования. Они помогают документировать такие атрибуты вариантов использования, как имена, приоритеты и стереотипы. В диаграммах прецедентов поддерживается несколько типов связей: связи коммуникации (Communication) – описывают связи между действующими лицами и вариантами использования; использования (uses) и расширения (extends) – отражают связи между вариантами использования; обобщения действующего лица (actor generalization) – между действующими лицами.

Связь использования позволяет одному варианту использования задействовать функциональность другого. С помощью таких связей обычно моделируют многократно применяемую функциональность, встречающуюся в двух или более вариантах использования.

Связь расширения позволяет варианту использования только при необходимости применять функциональные возможности другого варианта (extends).

На основе набора Use Case диаграмм создается список требований к системе и определяется множество выполняемых системой функций.

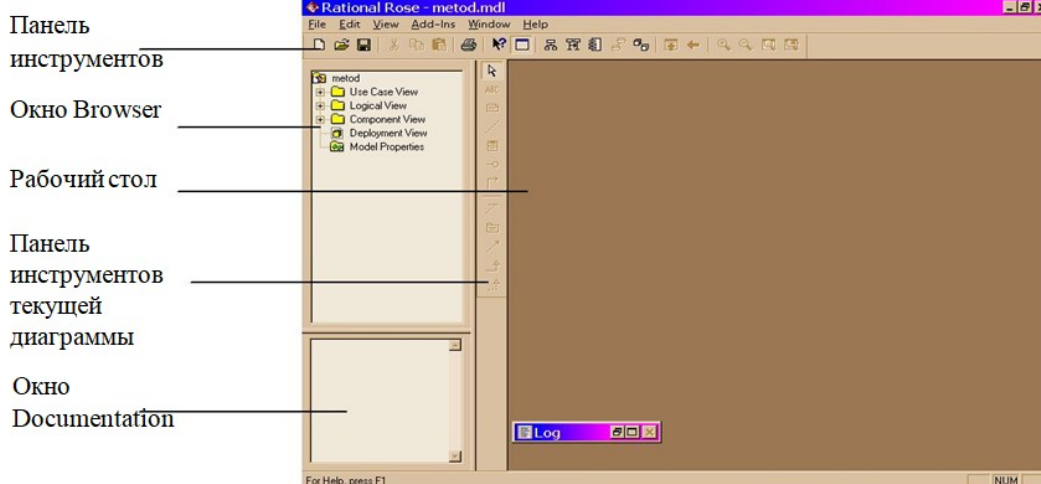


Рис.5 - Главное окно Rational Rose2000

В верхней части экрана, как у большинства редакторов в стиле Windows, находится меню и панель инструментов (Tool Bar).

Слева находится окно Browser для быстрого доступа к диаграммам. Это окно позволяет легко перемещаться по дереву диаграмм, буксировать диаграммы мышкой и изменять структуру модели по своему усмотрению.

Под окном Browser находится окно Documentation. В этом окне появляется описание, которое введено разработчиком для выделенного в текущий момент элемента.

В правой части экрана находятся те диаграммы, которые открыты в текущий момент, обычно это поле называется Рабочим столом Rational Rose.

Между окном Browser и окном Diagram находится панель инструментов текущей диаграммы, которая изменяется в зависимости от выбранной диаграммы.

Внизу рабочего стола находится свернутое окно Log (протокол). В нем постоянно фиксируются все действия, произведенные над диаграммами.

Построение диаграммы прецедентов

Откройте Главную диаграмму прецедентов (окно Browser > Use Case View > Main) или создайте новую диаграмму (.Use Case View > New > Use Case Diagram). (рис.6)

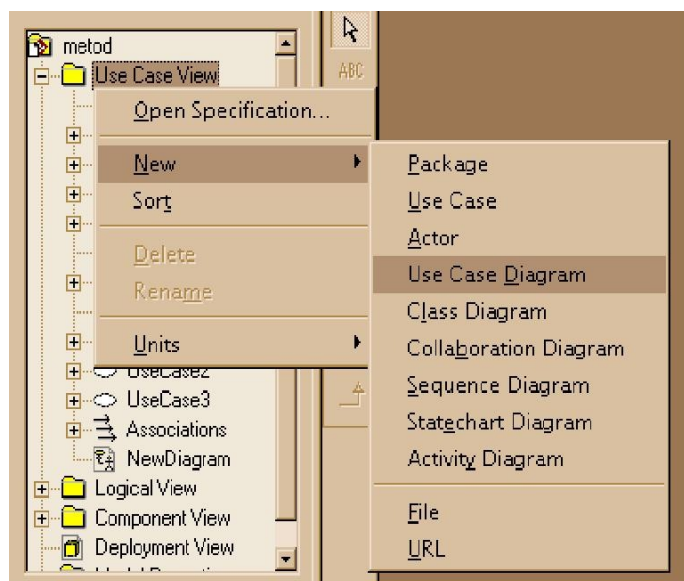


Рис.6

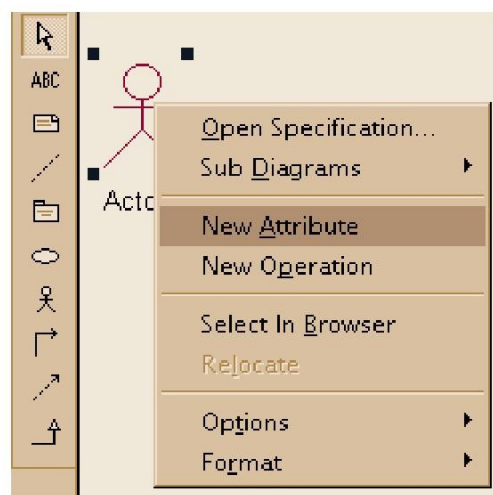


Рис.7

Создайте исполнителей, поместив их непосредственно на рабочий стол Rational Rose из строки инструментов текущей диаграммы, или выполнив последовательность: Use Case View>New>Actor. Атрибуты и операции можно задать из контекстного меню на рабочем столе или в окне броузера (New Attribute и New Operation). (рис.7)

Создайте прецеденты, поместив их непосредственно на рабочий стол из строки инструментов, или выполнив последовательность: Use Case View>New>Use Case.

Добавьте необходимые связи из строки инструментов. Для абстрактных исполнителей и абстрактных прецедентов используйте связи расширения (Generalization).

Задайте спецификации элементов (Open Specification в контекстном меню выбранного элемента). Для абстрактных исполнителей и прецедентов установите флажок Abstract (для исполнителей на вкладке Detail). На диаграмму можно поместить описания (Note), по необходимости соединив их с элементами пунктирной линией (Anchor Note to Item).

Контрольные вопросы

1. В чем состоит сущность объектно-ориентированного подхода?
2. Назовите основные понятия объектно-ориентированного подхода.
3. Какие принципы используются при построении общей модели в Rational Rose?
4. Назовите модели и диаграммы Rational Rose.
5. Что понимается под прецедентом в объектном моделировании?
6. Для чего строится диаграмма прецедентов?
7. Перечислите основные элементы диаграммы прецедентов.
8. Какие типы связей поддерживаются в диаграммах прецедентов?

Задания

Фирма «КИС Лтд» участвует в конкурсе на разработку лучшего проекта разработки и внедрения корпоративной информационной системы для управления сетью супермаркетов «Все сразу». Владельцы системы супермаркетов хотят получить сертификат соответствия системе качества ИСО 9000 (в частности, ИСО 9004-2 – предоставление услуг). Одним из этапов подготовки к получению сертификата системы качества является документирование всех бизнес процессов, происходящих на предприятии.

Разработать IDEF0, DFD, IDEF3 диаграммы бизнес процесса, указанного в варианте.

Номер варианта выбирается согласно номеру компьютера, за которым сидит исполнитель.

Варианты:

1. Принятие товара на склад
2. Отпуск товара со склада
3. Оформление договора с клиентом на оптовую продажу продукции
4. Оформление договора с поставщиком на оптовую поставку продукции
5. Операции банкомата
6. Обработка платежа с помощью пластиковой карты
7. Обработка платежа через кассовый аппарат с учетом дисконтных карточек
8. Принятие системы скидок в праздничные дни
9. Анализ работы кассового зала за один день
10. Работа с персоналом супермаркета
11. Принятие системы скидок на залежавшийся товар
12. Работа с арендаторами торговой площади

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №12

Тема: Диаграмма вариантов использования.

Цель : изучение объектно-ориентированного моделирования и исследование процесса построения диаграммы классов в заданной предметной области с помощью пакета Rational Rose 2000.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Диаграммы классов

На диаграммах классов отображаются некоторые классы и пакеты системы. Это статические картины фрагментов системы и связей между ними.

Обычно для описания системы создают несколько диаграмм классов. На одних показывают некоторое подмножество классов и отношения между классами подмножества. На других отображают то же подмножество, но вместе с атрибутами и операциями классов. Третьи соответствуют только пакетам классов и отношениям между ними. Для представления полной картины системы можно разработать столько диаграмм классов, сколько требуется.

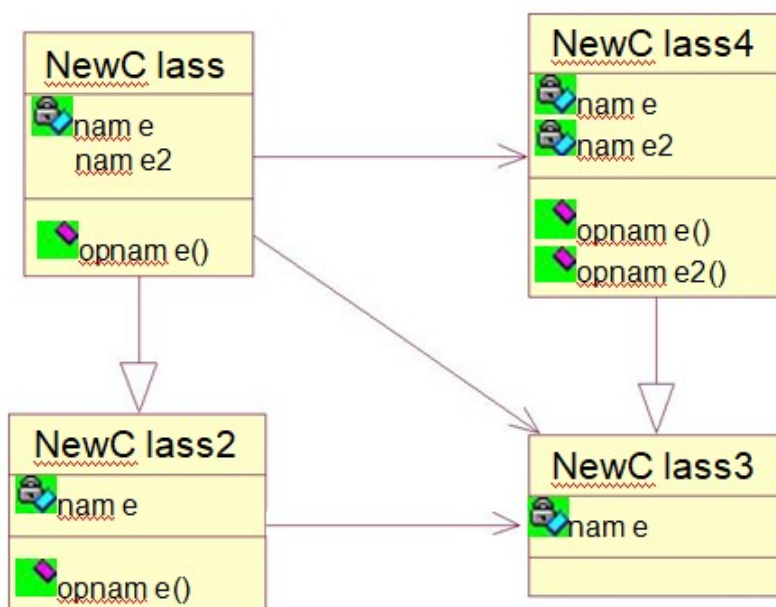


Рис.1 Диаграмма классов

По умолчанию существует одна диаграмма классов, называемая Главной (Main). На этой диаграмме показывают пакеты классов модели. Внутри каждого пакета также имеется Главная диаграмма, включающая в себя все классы этого пакета. На этой диаграмме показывают пакеты классов модели. Внутри каждого пакета также имеется Главная диаграмма, включающая в себя все классы этого пакета.

Выявление классов можно начать с изучения потока событий сценария. Имена существительные в описании этого потока дадут понять, что может являться классом.

Каждый объект диаграмм последовательности и кооперативных диаграмм должен быть соотнесен с соответствующим классом.

Поток событий и диаграммы взаимодействия являются прекрасным местом, где можно начать поиск классов.

Атрибут - это значение, характеризующее объект в его классе. Примеры атрибутов: категория, баланс, кредит (атрибуты объектов класса счет); имя, возраст, вес (атрибуты объектов класса человек) и т.д.

Среди атрибутов различаются постоянные атрибуты (константы) и переменные атрибуты. Постоянные атрибуты характеризуют объект в его классе (например, номер счета,

категория, имя человека и т.п.). Текущие значения переменных атрибутов характеризуют текущее состояние объекта (например, баланс счета, возраст человека и т.п.); изменяя значения этих атрибутов, мы изменяем состояние объекта.

Атрибуты перечисляются во второй части прямоугольника, изображающего класс (см. рисунок 1). Иногда указывается тип атрибутов (ведь каждый атрибут - это некоторое значение) и начальное значение переменных атрибутов (совокупность начальных значений этих атрибутов задает начальное состояние объекта).

Операция - это функция (или преобразование), которую можно применять к объектам данного класса. Одна и та же операция может, вообще говоря, применяться к объектам разных классов: такая операция называется полиморфной, так как она может иметь разные формы для разных классов.

Каждой операции соответствует метод - реализация этой операции для объектов данного класса. Таким образом, операция - это спецификация метода, метод - реализация операции. Например, в классе файл может быть определена операция печать (print). Эта операция может быть реализована разными методами: (а) печать двоичного файла; (б) печать текстового файла и др. Логически эти методы выполняют одну и ту же операцию, хотя реализуются они разными фрагментами кода.

Каждая операция имеет один неявный аргумент - объект к которому она применяется. Кроме того, операция может иметь и другие аргументы (параметры). Эти дополнительные аргументы параметризуют операцию, но не связаны с выбором метода. Метод связан только с классом и объектом (некоторые объектно-ориентированные языки, например С++, допускают одну и ту же операцию с разным числом аргументов, причем используя то или иное число аргументов, мы практически выбираем один из методов, связанных с такой операцией; на этапе предварительного проектирования системы удобнее считать эти операции различными, давая им разные имена, чтобы не усложнять проектирование).

Операция (и реализующие ее методы) определяется своей сигнатурой, которая включает, помимо имени операции, типы (классы) всех ее аргументов и тип (класс) результата (возвращаемого значения). Все методы, реализующие операцию должны иметь такую же сигнатуру, что и реализуемая ими операция.

Операции перечисляются в третьей части прямоугольника (рисунок 1), описывающего класс. Каждая операция должна быть представлена своей сигнатурой, однако на ранних стадиях проектирования можно ограничиваться указанием имени операции, отложив полное определение сигнатуры на конец рассматриваемой фазы жизненного цикла (либо даже на последующие фазы). В графическом языке технологии ОМТ тип любого объекта данных указывается вслед за именем этого объекта после двоеточия (как в языке Паскаль).

При моделировании системы полезно различать операции, имеющие побочные эффекты (эти эффекты выражаются в изменении значений атрибутов объекта, т.е. в изменении его состояния), и операции, которые выдают требуемое значение, не меняя состояния объекта. Эти последние операции называются запросами.

Таким образом, для задания класса необходимо указать имя этого класса, а затем перечислить его атрибуты и операции (или методы). Полное описание объекта на графическом языке ОМТ имеет вид, изображенный на рисунке 2. Однако иногда удобно бывает пользоваться сокращенным описанием класса, когда в прямоугольнике, изображающем этот класс, указывается только имя класса.

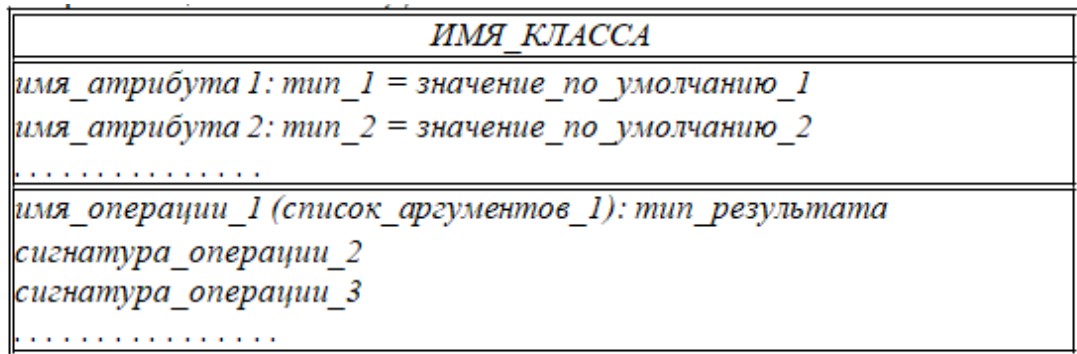


Рис. 2. Полное представление объекта в ОМТ

Параметризованный класс (parameterized class) — один из специальных типов классов. Он применяется для создания семейства других классов. Обычно параметризованный класс является разновидностью контейнера, его еще называют шаблоном.

На языке UML параметризованный класс изображается с помощью следующей нотации:



Рис.3

Для добавления параметризованного класса нажмите кнопку Parameterized Class (Параметризованный класс) панели инструментов, щелкните мышью где-нибудь на диаграмме, чтобы поместить туда новый класс.

Класс-наполнитель (instantiated class) является параметризованным классом, аргументы которого имеют фактические значения. В соответствии с нотацией UML, имя аргумента класса-наполнителя заключается в угловые скобки (< >).

Для добавления класса-наполнителя нажмите кнопку Instantiated Class панели инструментов, щелкните мышью на диаграмме там, где будет располагаться новый класс. Введите имя класса с аргументами, заключив их в угловые скобки. В окне спецификации класса в поле Type (Тип) укажите InstantiatedClass.

Утилита класса (class utility) — это совокупность операций. Например, это может быть совокупность математических функций, которые используются всей системой и не слишком хорошо подходят для какого-либо конкретного класса. Эти функции можно собрать вместе и объединить в утилиту класса, которая будет использоваться другими классами системы.

Утилиты классов часто применяют для расширения функциональных возможностей языка программирования или для хранения общих элементов функциональности многократного использования, необходимых в нескольких системах.

Утилита класса выглядит на диаграмме как класс "с тенью"

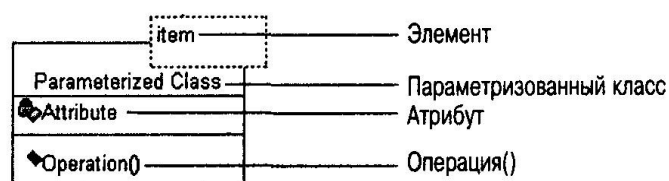


Рис.4

Утилитой параметризованного класса (parameterized class utility) является параметризованный класс, содержащий только набор операций. Это шаблон для создания утилит класса. На диаграмме классов она выглядит следующим образом:

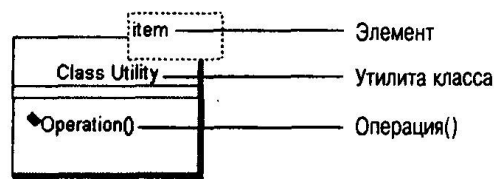


Рис.5

Метакласс (metaclass) — это класс, экземпляры которого являются классами, а не объектами. К числу метаклассов относятся параметризованные классы и утилиты параметризованных классов. На языке UML метаклассы изображают следующим образом:

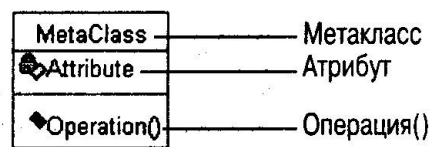


Рис.6

Для добавления метакласса добавьте класс на диаграмму Классов или в браузер, в спецификации класса в поле Type (Тип) укажите MetaClass (Метакласс).

Каждому классу модели Rose необходимо дать уникальное имя. В общем случае используются существительные в единственном числе.

Обычно имена классов не содержат пробелов. Это делается по практическим причинам и из соображений удобочитаемости — языки программирования, как правило, не поддерживают пробелы в именах классов.

Стереотип — это механизм, позволяющий категоризировать классы. Допустим, вы хотите найти все формы в вашей модели. Для этого можно создать стереотип Form (Форма) и назначить его всем окнам вашего приложения. В дальнейшем при поиске форм нужно только искать классы с этим стереотипом. На языке UML определены три основных стереотипа: Boundary (Граница), Entity (Объект) и Control (Управление).

Пограничными классами (boundary classes) называются такие классы, которые расположены на границе системы со всем остальным миром. Они включают в себя формы, отчеты, интерфейсы с аппаратурой такой, как принтеры или сканеры) и интерфейсы с другими системами.

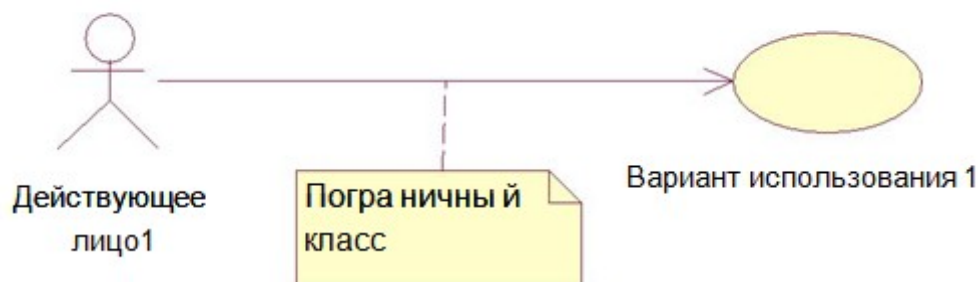


Рис.7

Для выявления пограничных классов необходимо исследовать диаграммы Вариантов Использования. Для каждого взаимодействия между действующим лицом и вариантом

использования должен существовать хотя бы один пограничный класс. Именно он позволяет действующему лицу взаимодействовать с системой.

Классы-сущности (entity classes) содержат информацию, хранимую постоянно. Классы-сущности можно обнаружить в потоке событий и на диаграммах Взаимодействия. Они имеют наибольшее значение для пользователя, и потому в их названиях часто применяют термины из предметной области.

Часто для каждого класса-сущности создают таблицу в базе данных. В этом заключается еще одно отличие от типичного подхода к конструированию системы. Вместо того чтобы с самого начала задавать структуру базы данных, у нас теперь есть возможность разрабатывать ее на основе информации получаемой из объектной модели. Это позволяет отслеживать соответствие всех полей базы данных и требований к системе. Требования определяют поток событий. Поток событий определяет классы и атрибуты классов. Каждый атрибут класса-сущности становится полем в базе данных. Применяя такой подход, легко отслеживать соответствие между полями базы данных и требованиями к системе, что уменьшает вероятность сбора ненужной информации.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №13

Тема: Изучение программного пакета Rational Rose 2000. Диаграмма классов

Цель : изучение объектно-ориентированного моделирования и исследование процесса построения диаграммы классов в заданной предметной области с помощью пакета Rational Rose 2000.

Управляющие классы (control classes) отвечают за координацию действий других классов. Обычно у каждого варианта использования имеется один управляющий класс, контролирующий последовательность событий этого варианта использования.

Сам управляющий класс не несет в себе никакой функциональности -остальные классы посылают ему мало сообщений. Но сам он посылает множество сообщений. Управляющий класс делегирует ответственность другим классам. По этой причине управляющий класс часто называют классом-менеджером. Параметр Visibility (Видимость) показывает, будет ли класс виден вне своего пакета:

1. Public (открытый) - этот класс виден всем остальным классам системы.
2. Protected, Private (защищенный, закрытый) - класс может быть виден во вложенных в него классах, "друзьям" (friends) этого класса или из самого класса.
3. Package or Implementation (пакет или реализация) - класс может быть виден только из классов того же пакета.

Для установки видимости класса в спецификации установите параметр Export control (Контроль экспорта) в значение Public; Protected, Private или Implementation.

Поле Cardinality (Множественность) позволяет указать, сколько у данного класса должно быть экземпляров.

Множественность управляющего класса обычно равна 1. В Rose доступны следующие значения множественности:

- N – много (по умолчанию)
- 0..0 – нуль
- 0..1 – нуль или один – 0..n – нуль или больше – 1..1 – ровно один
- 1..n – один или больше

Можно вводить множественность самим:

- <число> - ровно число
- <число1>..<число2> - между числом 1 и числом 2 – <число>..n – число или больше
- <число1>,<число2> - число1 или число2

– <число1>,<число2>..<число3> - ровно число 1 или между числом 2 и 3 –
<число1>..<число2>,<число3>..<число4> - между 1 и 2 или между 3 и 4

Для задания множественности класса в спецификации на вкладке Detail (Подробно) укажите множественность в раскрывающемся списке или введите ее значение сами.

Абстрактным называется класс, который не наполняется конкретным содержанием. Иными словами, если класс А абстрактный, в памяти никогда не будет объектов типа А.

Обычно абстрактные классы применяют при работе с наследованием. В них содержатся данные и поведение, общие для нескольких других классов.

В Rose классы можно вкладывать друг в друга. Во вложенные (nested) классы можно вкладывать другие классы, организуя столько уровней вложения, сколько необходимо. Для создания вложенного класса на вкладку Nested спецификации родительского класса щелкните правой кнопкой мыши на белом поле этой вкладки. В открывшемся меню выберите пункт Insert. Введите имя вложенного класса.

Если нужно показать вложенный класс на диаграмме Классов выберите в меню модели пункт Query →-Add Classes. Перетащите вложенный класс из списка Classes в список Selected Classes.

Пакеты (packages) применяются для группирования классов, обладающих некоторой общностью. Объединять классы можно, как угодно, однако существует несколько наиболее распространенных

подходов. Во-первых, можно группировать классы по стереотипу. В таком случае получается один пакет с классами-сущностями, один с пограничными классами, один с управляющими классами и т.д.

Второй подход заключается в объединении классов по их функциональности. Наконец, применяют комбинацию двух указанных подходов.

Поведение системы обеспечивается взаимодействием объектов. Два типа отношений, которые можно выделить на этапе анализа – это ассоциация и агрегация.

Ассоциация (association) – это двунаправленная семантическая связь между классами.

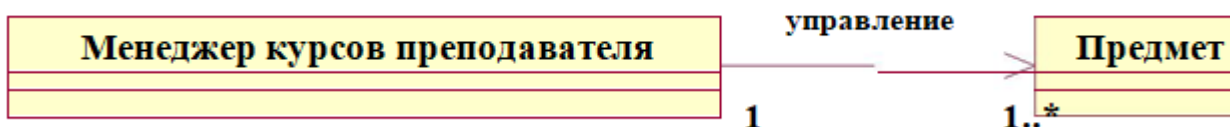


Рис.8

Агрегационное отношение – это специальная форма ассоциации между целым и его частью или частями, то есть отношение типа «часть от» или «содержит».



Рис.9

Мощность отношений (multiplicity) указывается для классов и определяет допустимое количество объектов, участвующих в отношении с каждой стороны.

Несколько объектов, принадлежащих одному классу, могут взаимодействовать друг с другом. Такое взаимодействие показывается на диаграмме классов как возвратное.

Наследованием называется такое отношение между классами, когда один класс использует часть структуры и / или поведения другого или нескольких классов. При наследовании создается иерархия абстракций, в которой подкласс (subclass) наследуется от

одного или нескольких суперклассов (superclass). Подкласс наследует все атрибуты, операции и отношения, определенные в каждом его суперклассе.

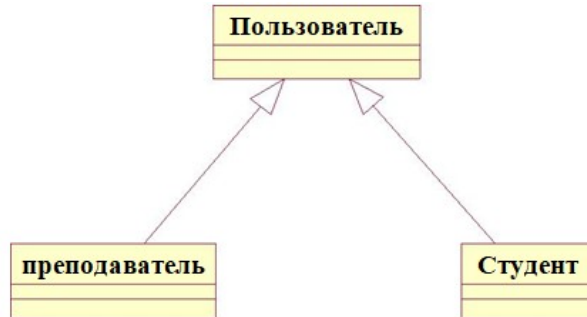


Рис.10

Диаграммы Классов являются хорошим инструментом проектирования.

С их помощью разработчики могут видеть и планировать структуру системы еще до фактического написания кода, благодаря чему они с самого начала могут понять, хорошо ли спроектирована система.

Практическое занятие №14

Тема: Изучение программного пакета Rational Rose 2000.

Цель : изучение объектно-ориентированного моделирования и исследование процесса построения диаграммы классов в заданной предметной области с помощью пакета Rational Rose 2000.

Построение диаграммы классов

1. Откройте Главную диаграмму классов (окно Browser > Logical View > Main) или создайте новую диаграмму (.Logical View > New > Class Diagram). (рис.11)

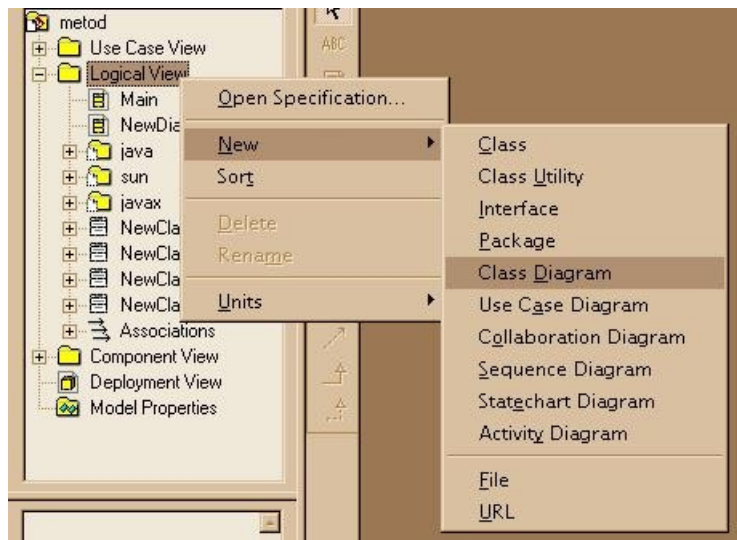


Рис.11

2. Для того чтобы создать пакеты, переносят их непосредственно на рабочий стол Rational Rose из строки инструментов текущей диаграммы, или выполняют последовательность: Logical View > New > Package. Пакеты соединяют стрелками, если необходимо показать их связи.

3. Внутри каждого пакета можно создать вложенную диаграмму классов. Для этого можно щелкнуть мышкой на значке пакета на рабочем столе или в окне браузера выполнить последовательность

<Package>>New> Class Diagram.

4. Для того чтобы создать классы, переносят их непосредственно на рабочий стол Rational Rose из строки инструментов текущей диаграммы, выполняют последовательность: Logical View>New>Class (для общих классов), создают класс в конкретном пакете (<Package> >New>Class) или перетаскивают уже существующий класс из окна браузера на рабочий стол.

5. Спецификации, атрибуты и операции классов можно задать из контекстного меню на рабочем столе или в окне браузера (Open Specification, New Attribute и New Operation).

6. Задайте связи между классами.

Порядок выполнения

1. Постройте диаграмму классов или, если требуется, несколько диаграмм.
2. Ответьте на контрольные вопросы.

Содержание отчета

1. Название работы.
2. Цель работы.
3. Концептуальная диаграмма классов.
4. Детализированная диаграмма классов.
5. Конкретизированная диаграмма классов.
6. Описание диаграмма.

Контрольные вопросы

1. Что такое объект в объектном моделировании?
2. Что понимается под классом в объектном моделировании?
3. Какие типы отношений имеются в нотации UML?
4. Назовите основные стереотипы класса.
5. Что такое диаграмма классов?
6. Назовите основные элементы диаграммы классов.
7. Назовите основную характеристику класса.
8. Что понимается под операцией в диаграмме классов?
9. Назовите основные типы классов.
10. Какие типы связей поддерживаются в диаграммах классов?

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №15

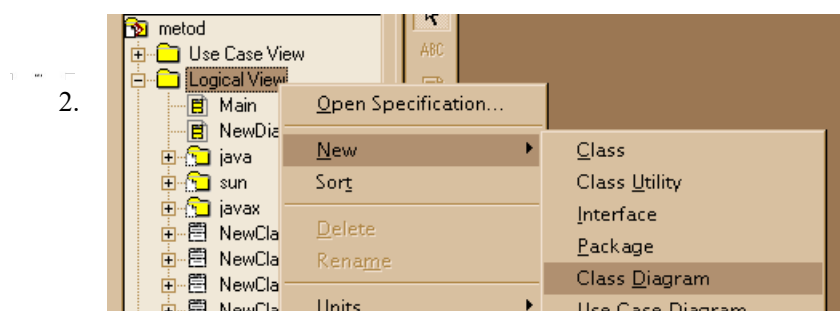
Тема: Диаграмма классов

Цель : изучение объектно-ориентированного проектирования и исследование процесса построения диаграмм последовательности для моделирования бизнес-процессов в предметной области с помощью пакета Rational Rose 2000.

Диаграммы Классов являются хорошим инструментом проектирования.

С их помощью разработчики могут видеть и планировать структуру системы еще до фактического написания кода, благодаря чему они с самого начала могут понять, хорошо ли спроектирована система.

1. Откройте Главную диаграмму классов (окно Browser > Logical View >Main) или создайте новую диаграмму (.Logical View >New>Class Diagram). (рис.11)



Для того чтобы создать пакеты, переносят их непосредственно на рабочий стол Rational Rose из строки инструментов текущей диаграммы,



age.Пакеты соединяют стрелками,

3. Внутри каждого пакета можно создать вложенную диаграмму классов. Для этого можно щелкнуть мышкой на значке пакета на рабочем столе или в окне браузера выполнить последовательность <Package>>New> Class Diagram.
4. Для того чтобы создать классы, переносят их непосредственно на рабочий стол Rational Rose из строки инструментов текущей диаграммы, выполняют последовательность: Logical View>New>Class (для общих классов), создают класс в конкретном пакете (<Package>>New>Class) или перетаскивают уже существующий класс из окна браузера на рабочий стол.
5. Спецификации, атрибуты и операции классов можно задать из контекстного меню на рабочем столе или в окне браузера (Open Specification, New Attribute и New Operation).
6. Задайте связи между классами.

Порядок выполнения лабораторной работы

1. Постройте диаграмму классов или, если требуется, несколько диаграмм.
2. Ответьте на контрольные вопросы.

Контрольные вопросы

1. Что такое объект в объектном моделировании?
2. Что понимается под классом в объектном моделировании?
3. Какие типы отношений имеются в нотации UML?
4. Назовите основные стереотипы класса.
5. Что такое диаграмма классов?
6. Назовите основные элементы диаграммы классов.
7. Назовите основную характеристику класса.
8. Что понимается под операцией в диаграмме классов?
9. Назовите основные типы классов.
10. Какие типы связей поддерживаются в диаграммах классов?.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №16

Тема: Изучение программного пакета Rational Rose 2000. Диаграммы деятельности

Цель : изучение объектно-ориентированного проектирования и исследование процесса построения диаграмм последовательности для моделирования бизнес-процессов в предметной области с помощью пакета Rational Rose 2000.

Диаграммы деятельности

Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого является состояние действия, а дугами – переходы от одного состояния в действия к другому.

При этом каждое состояние может являться выполнением операции некоторого класса либо её часть, позволяя использовать диаграмму деятельности для описания реакции на внутренние события системы.

На рис. представлен алгоритм процесса принятия решения при настройке и внедрении программного продукта, описанный с помощью диаграммы действий.

В языке UML действие изображается в виде прямоугольника с закругленными углами, состояния - в виде прямоугольника, переходы – в виде направленных стрелок, элементы выбора – в виде ромбов, линии синхронизации – в виде толстых горизонтальных или вертикальных линий.

Состоянием называется одно из возможных условий, в которых может существовать объект. Для выявления состояний объекта необходимо исследовать две области модели: значения атрибутов объекта и связи с другими объектами. В рассматриваемом примере: "Сформированы требования к ПП", "Выданы рекомендации".

Существуют специальные состояния объекта: начальное и конечное. Начальным (Start State) называется состояние, в котором находится объект сразу после своего создания (В примере это "заказ на ПО"). Конечным (End State) называется состояние, в котором объект находится непосредственно перед уничтожением ("Решение по адаптации").

Действием называется исполнение определенного поведения в потоке управления системы. В примере это "Поиск правила", "Формировать решение" и т.д.

Переходы используются для изображения пути потока управления от действия к действию.

Элементы выбора показывают места разделения управляющих потоков на основе условного выбора. Переходы из элементов выбора содержат ограничительные условия, определяющие, какое направление перехода будет выбрано ("Поиск правила", "Поиск прецедента").

Линии синхронизации (synhronization bar) позволяет указать на необходимость одновременного выполнения действий, а также обеспечивает единое выполнение действий в потоке ("Формировать решение", "Адаптировать решение").

Секции делят диаграмму действий на несколько участков, чтобы показать, кто отвечает за выполнение действий на каждом участке.

Диаграммы действий отражают динамику проекта и представляют собой схемы потоков управления в системе от действия к действию, а также параллельные действия и альтернативные потоки.

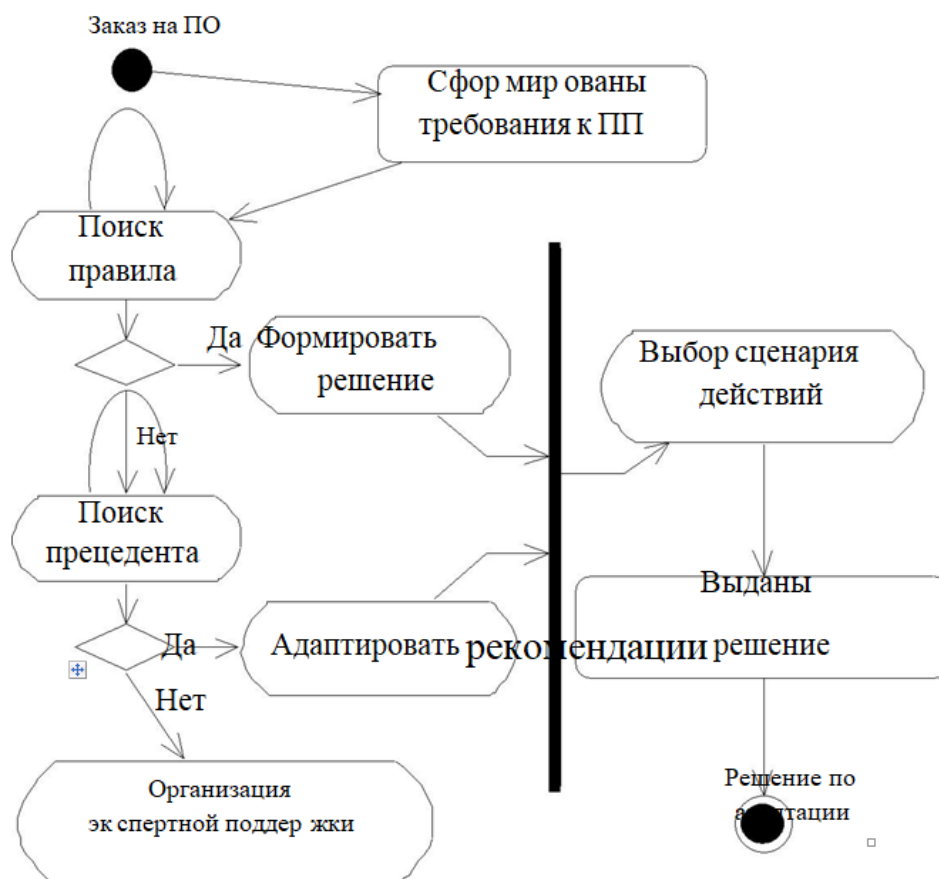


Рис - Диаграмма действий

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №17

Тема: Изучение объектно-ориентированного проектирования и исследование процесса построения диаграмм последовательности для моделирования бизнес-процессов в предметной области с помощью пакета Rational Rose 2000.

Цель : изучение объектно-ориентированного проектирования и исследование процесса построения диаграмм последовательности для моделирования бизнес-процессов в предметной области с помощью пакета Rational Rose 2000.

Rational Rose

Rational Rose представляет собой CASE средство проектирования и разработки информационных систем и программного обеспечения для управления предприятиями. Как и другие CASE средства (ARIS, BPwin, ERwin) его можно применять для анализа и моделирования бизнес процессов. Первая версия этого продукта была выпущена компанией Rational Software . В дальнейшем Rational Rose был куплен IBM.

Принципиальное отличие Rational Rose от других средств заключается в объектно-ориентированном подходе. Графические модели, создаваемые с помощью этого средства, основаны на объектно-ориентированных принципах и языке UML (Unified Modeling Language). Инструменты моделирования Rational Rose позволяют разработчикам создавать целостную архитектуру процессов предприятия, сохраняя все взаимосвязи и управляющие воздействия между различными уровнями иерархии.

Моделирование бизнес процессов в Rational Rose выполняется за счет применения различных аспектов. Каждый из этих аспектов концентрирует внимание на определенных характеристиках и возможностях процессов.

К таким аспектам относятся:

Вариант использования (Use case). Этот аспект дает возможность понять, каким



образом действуют участники процесса и за счет этого определить их взаимодействие и влияние на процесс. Для построения моделей процесса в рамках данного аспекта применяются Use-case диаграммы, диаграммы последовательностей, диаграммы совместной работы и диаграммы действий.

Логический аспект. С помощью этого аспекта можно определить функциональные требования процессов. Он задает логическую взаимосвязь между классами элементов процессов. Для построения моделей применяются диаграммы классов и диаграммы состояний.

Составляющие элементы. Этот аспект обращает внимание на состав элементов

процесса и их распределение при создании информационной системы. Модели в этом аспекте строятся с помощью диаграммы компонентов. Она содержит информацию об элементах процесса и программном обеспечении.

Ввод в действие. Этот аспект показывает схему процесса в привязке к аппаратному обеспечению информационной системы. Для построения моделей применяется только одна диаграмма – диаграмма топологии.

За счет применения различных аспектов Rational Rose предоставляет пользователям (бизнес аналитикам, инженерам, техническим специалистам и руководителям) возможность создавать, анализировать, изменять и управлять моделями, используя единый объектно-ориентированный подход и единый язык моделирования. Возможности Rational Rose

Последние версии Rational Rose содержат несколько программных продуктов, которые обеспечивают широкие возможности по моделированию бизнес процессов. Пользователи могут создавать графические модели процессов, приближенные к потребностям бизнеса.

Rational Rose обеспечивает следующие возможности моделирования бизнес процессов:

Поддержка объектного моделирования. Применение принципов объектного моделирования и языка UML позволяет приблизить модели процессов к требованиям бизнеса и упрощает вид моделей.

Структурное представление элементов. Модели процессов и их элементы могут быть представлены в виде графической структуры, наглядно отображающий их состав и взаимосвязи.

Интеграция моделей. За счет применения единого языка UML, Rational Rose позволяет объединить модели бизнес процесса, модели приложений и модели данных.

Интеграция с программными продуктами. Для расширения возможностей моделирования и анализа бизнес процессов в Rational Rose реализована возможность интеграции с другими программными продуктами, например, Microsoft Visual Studio.

Открытая архитектура. Она позволяет дополнять существующий инструментарий программы новыми функциями и возможностями.

Обратное проектирование. Эта возможность позволяет на основе имеющегося программного кода построить понятийную модель. Для целей моделирования бизнес процессов данная возможность может быть полезна, если моделируемый процесс автоматизирован.

Преимущества Rational Rose

Основное преимущество данного CASE средства связано с объектным принципом моделирования. За счет его применения можно максимально сблизить представления различных специалистов, которые осуществляют моделирование бизнес процессов и работают с моделями. Помимо этого, есть и преимущества, обусловленные удобством работы с программным пакетом Rational Rose .

Преимущества Rational Rose являются:

Поддержка командной работы. В этом CASE средстве реализована простая поддержка всех участников проекта. Пользователи могут работать со своими собственными уникальными моделями и в своем собственном окружении без смены рабочего места, при этом сохраняется взаимосвязь с общими моделями.

Управление моделями. Все создаваемые модели могут быть легко изменены. Изменения в одной модели автоматически отражаются во взаимосвязанных моделях. Для управления моделями применяется система контроля версий и управления конфигурацией. Это позволяет легко проводить изменения в любых моделях бизнес процессов.

Контроль ошибок. Rational Rose обеспечивает отслеживание ошибок, возникающих при моделировании. Это позволяет исправить ошибки с учетом их наследования и передачи на очередной уровень моделирования.

Документирование моделей. Пользователи могут создавать необходимые им отчеты и документы по моделям процессов. Документы формируются под потребности пользователя и могут настраиваться для применения к разным моделям.

Управление конфигурацией. Пользователи могут настраивать конфигурацию интерфейса и части приложений под свои потребности. В Rational Rose применяется графический пользовательский интерфейс (GUI), за счет которого можно настроить необходимое окружение для комфортной работы.

Практическое занятие №18

Тема: Диаграмма деятельности

Цель : изучение объектно-ориентированного проектирования и исследование процесса построения диаграмм последовательности для моделирования бизнес-процессов в предметной области с помощью пакета Rational Rose 2000.

Построение диаграммы деятельности

1. Создайте диаграмму действий (.New> Activity Diagram).
2. Создайте необходимые элементы (состояния, действия, условия, связи) поместив их на рабочий стол из строки инструментов. Не забудьте про начальные и конечные состояния.

Порядок выполнения лабораторной работы

1. Постройте диаграмму деятельности для двух-трех процессов. Одну из диаграмм построить с разбивкой на дорожки.
2. Ответьте на контрольные вопросы.

Содержание отчета

1. Название работы.
2. Цель работы.
 1. Диаграммы деятельности с необходимыми комментариями.

Контрольные вопросы

1. Для чего строятся диаграммы действия?
2. Перечислите основные элементы диаграммы действий.
3. Что такое «состояние»? Назовите специальные виды состояний.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №19

Тема: Изучение программного пакета Rational Rose 2000.

Цель: изучение объектно-ориентированного моделирования и исследование процесса построения диаграммы состояний и последовательности в заданной предметной области с помощью пакета Rational Rose 2000.

Диаграмма состояний

Диаграмма состояний (statechart diagram) показывает положение одиночного объекта, события или сообщения, которые вызывают переходы из одного состояния в другое, и действия, являющиеся результатом смены состояний.

На диаграмме состояний отображают жизненный цикл одного объекта, начиная с момента его создания и заканчивая разрушением. С помощью таких диаграмм удобно моделировать динамику поведения класса. Как правило, диаграммы состояний не требуется создавать для каждого класса. Диаграмма состояний создается для классов с динамическим поведением. Многие проекты вообще обходятся без них.

Состояние – это некоторое положение в жизни объекта, при котором он удовлетворяет определенному условию, выполняет некоторое действие или ожидает события.

Диаграмма состояний включает все сообщения, которые объект получает и отправляет. Сценарий – это одиночный проход по диаграмме состояний. Интервал между

двумя сообщениями, отправляемыми объектом. Обычно представляет состояние. Конечный автомат представляет собой глубокий взгляд на поведение конкретного объекта. Спецификация автомата настолько подробна, что годится для непосредственной реализации в программном коде. Однако конечные автоматы описывают только единичные объекты, а не их совокупность.

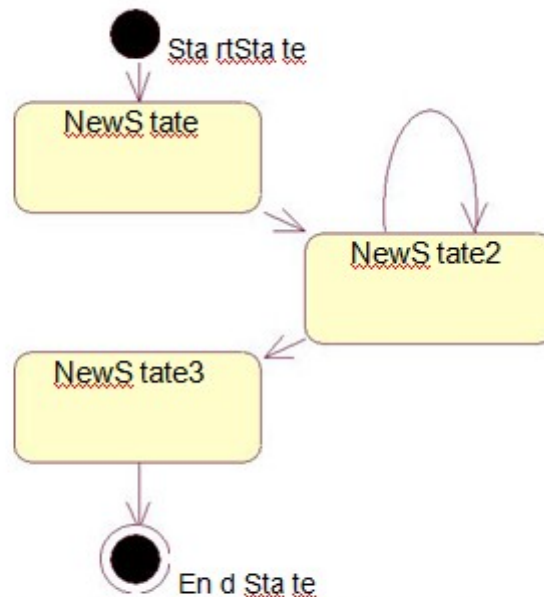


Рис.1 – Диаграмма состояний

Переходы между состояниями представляют собой смену исходного состояния последующим (которое может быть тем же, что и исходное). Переход может сопровождаться определенным действием.

С переходом между состояниями может быть связано условие (guard condition) и/или определенное действие (action). Переход может также вызывать событие (event). Действие – это поведение, проявляющееся при возникновении перехода. Событие – это сообщение, отправляемое другому объекту системы. Условие – булево выражение значений атрибутов, которое допускает переход, только если оно верно. И действие, и проверка условия представляют собой поведение объекта и обычно реализуются в виде операций.

Действия, сопровождающие переходы в определенное состояние, можно рассматривать как входные действия (entry action) для этого состояния. И наоборот, действия, сопровождающие переходы из данного состояния, являются для него выходными (exit action). Поведение, возникающее внутри состояния, называется деятельностью (activity). Деятельность начинается при входе в состояние и завершается или прерывается при выходе из него.

Переходом (transition) называется перемещение из одного состояния в другое. Совокупность переходов диаграммы показывает, как объект может переходить из одного состояния в другое. На диаграмме переходы изображают в виде стрелки, начинающейся в начальном и заканчивающейся в последующем состоянии. Переходы могут быть рефлексивными: объект переходит в то же состояние, в котором он в настоящий момент находится. Рефлексивные переходы изображают в виде стрелки, начинающейся и завершающейся на одном и том же состоянии.

Для уменьшения беспорядка на диаграмме можно вкладывать состояния одно в другое. Вложенные состояния называются подсостояниями (substates), а те, в которые они вложены, — суперсостояниями (superstates).

Если у нескольких состояний имеются идентичные переходы, эти состояния можно сгруппировать вместе в суперсостояние. Затем, вместо того чтобы поддерживать одинаковые переходы (по одному на каждое состояние), их можно объединить, перенеся в суперсостояние.

Суперсостояния позволяют навести порядок на диаграмме Состояний.

Для анализа динамики поведения класса необходимо рассмотреть его атрибуты. Экземпляр класса может вести себя по-разному в зависимости от их значений.

Полезно исследовать связи между классами. Рассмотрите все связи, множественность которых может принимать нулевое значение. Нули указывают, что данная связь не является обязательной. Ведет ли себя экземпляр класса по-разному при наличии и отсутствии связи? Если да, то он имеет несколько состояний.

В среде Rose на основании диаграмм Состояний не генерируется никакого исходного кода. Они нужны для того, чтобы документировать динамику поведения класса, благодаря чему аналитики и разработчики получают о нем четкое представление. Реализацией заложенной в эти диаграммы логики будут заниматься разработчики. Как и в случае других диаграмм UML, диаграммы Состояний дают возможность обсудить и документировать логику перед началом процесса кодирования.

Диаграммы последовательности

Объекты

На диаграмме последовательности изображаются только те объекты, которые непосредственно участвуют во взаимодействии. Ключевым моментом для диаграмм последовательности является динамика взаимодействия объектов во времени.

В UML диаграмма последовательности имеет как бы два измерения. Первое слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии. Крайним слева на диаграмме изображается объект, который является инициатором взаимодействия. Правее изображается другой объект, который непосредственно взаимодействует с первым. Таким образом, все объекты на диаграмме последовательности образуют некоторый порядок, определяемый очередностью или степенью активности объектов при взаимодействии друг с другом.

Графически каждый объект изображается прямоугольником и располагается в верхней части своей линии жизни. Внутри прямоугольника записываются имя объекта и имя класса разделенные двоеточием. При этом вся запись подчеркивается, что является признаком объекта.

Вторым измерением диаграммы последовательности является вертикальная временная ось, направленная сверху вниз. Начальному моменту времени соответствует самая верхняя часть диаграммы. Взаимодействия объектов реализуются посредством сообщений, которые посылаются одними объектами другим. Сообщения изображаются в виде горизонтальных стрелок с именем сообщения, а их порядок определяется временем возникновения. То есть, сообщения, расположенные на диаграмме последовательности выше, инициируются раньше тех, которые расположены ниже. Масштаб на оси времени не указывается, поскольку диаграмма последовательности моделирует лишь временную упорядоченность взаимодействий типа «раньше-позже».

Линия жизни объекта

Линия жизни объекта (object lifeline) изображается пунктирной вертикальной линией, ассоциированной с единственным объектом на диаграмме последовательности. Линия жизни служит для обозначения периода времени, в течение которого объект существует в системе и, следовательно, может потенциально участвовать во всех ее взаимодействиях. Если объект

существует в системе постоянно, то и его линия жизни должна продолжаться по всей плоскости диаграммы последовательности от самой верхней ее части до самой нижней.

Отдельные объекты, выполнив свою роль в системе, могут быть уничтожены, чтобы освободить занимаемые ими ресурсы. Для таких объектов линия жизни обрывается в момент его уничтожения. Для обозначения момента уничтожения объекта в языке UML используется специальный символ в форме латинской буквы «X». Ниже этого символа пунктирная линия не изображается, поскольку соответствующего объекта в системе уже нет, и этот объект должен быть исключен из всех последующих взаимодействий.

Не обязательно создавать все объекты диаграммы в начальный момент времени. Отдельные объекты могут создаваться по мере необходимости, экономя ресурсы системы и повышая ее производительность. В этом случае прямоугольник объекта изображается не в верхней части диаграммы последовательности, а в той части, которая соответствует моменту создания объекта. При этом прямоугольник объекта вертикально располагается в том месте диаграммы, которое по оси времени совпадает с моментом его возникновения в системе. Объект обязательно создается со своей линией жизни и, возможно, с фокусом управления.

Фокус управления

В процессе функционирования объектно-ориентированных систем одни объекты могут находиться в активном состоянии, непосредственно выполняя определенные действия, или состоянии пассивного ожидания сообщений от других объектов. Чтобы явно выделить подобную активность объектов, в языке UML применяется специальное понятие, получившее название фокуса управления (focus of control). Фокус управления изображается в форме вытянутого узкого прямоугольника, верхняя сторона которого обозначает начало получения фокуса управления объектом (начало активности), а его нижняя сторона - окончание фокуса управления (окончание активности). Прямоугольник располагается ниже обозначения соответствующего объекта и может заменять его линию жизни, если на всем ее протяжении он является активным.

Периоды активности объекта могут чередоваться с периодами его пассивности или ожидания. В этом случае у такого объекта имеются несколько фокусов управления. Важно сознавать, что получить фокус управления может только существующий объект, у которого в этот момент имеется линия жизни. Если же некоторый объект был уничтожен, то вновь возникнуть в системе он уже не может. Вместо него лишь может быть создан другой экземпляр этого же класса, который, строго говоря, будет являться другим объектом.

В отдельных случаях инициатором взаимодействия в системе может быть актер или внешний пользователь. В этом случае актер изображается на диаграмме последовательности самым первым объектом слева со своим фокусом управления. Чаще всего актер и его фокус управления будут существовать в системе постоянно, отмечая характерную для пользователя активность в инициировании взаимодействий с системой. При этом актер может иметь собственное имя или оставаться анонимным.

Иногда некоторый объект может инициировать рекурсивное взаимодействие с самим собой. Наличие во многих языках программирования специальных средств построения рекурсивных процедур требует визуализации соответствующих понятий в форме графических примитивов. На диаграмме последовательности рекурсия обозначается небольшим прямоугольником, присоединенным к правой стороне фокуса управления того объекта, для которого изображается это рекурсивное взаимодействие.

Сообщения

В UML каждое взаимодействие описывается совокупностью сообщений, которыми участвующие в нем объекты обмениваются между собой. Сообщение (message) представляет собой законченный фрагмент информации, который отправляется одним

объектом другому. Прием сообщения инициирует выполнение определенных действий, направленных на решение отдельной задачи тем объектом, которому это сообщение отправлено.

Таким образом, сообщения не только передают некоторую информацию, но и требуют или предполагают выполнения ожидаемых действий от принимающего объекта. Сообщения могут инициировать выполнение операций объектом соответствующего класса, а параметры этих операций передаются вместе с сообщением. На диаграмме последовательности все сообщения упорядочены по времени своего возникновения

В моделируемой системе. В таком контексте каждое сообщение имеет направление от объекта, который инициирует и отправляет сообщение, к объекту, который его получает. Иногда отправителя сообщения называют клиентом, а получателя сервером. Тогда сообщение от клиента имеет форму запроса некоторого сервиса, а реакция сервера на запрос после получения сообщения может быть связана с выполнением определенных действий или передачи клиенту необходимой информации тоже в форме сообщения.

Сообщения изображаются горизонтальными стрелками, соединяющими линии жизни или фокусы управления двух объектов на диаграмме последовательности. В языке UML различаются несколько разновидностей сообщений, каждое из которых имеет свое графическое изображение:

первая разновидность сообщения является наиболее распространенной и используется для вызова процедур, выполнения операций или обозначения отдельных вложенных потоков управления. Начало этой стрелки всегда соприкасается с фокусом управления или линией жизни того объекта-клиента, который инициирует это сообщение. Конец стрелки соприкасается с линией жизни того объекта, который принимает это сообщение и выполняет в ответ определенные действия. Принимающий объект, как правило, получает фокус управления, становясь активным;

вторая разновидность сообщения используется для обозначения простого потока управления. Каждая такая стрелка указывает на выполнение одного шага потока. Такие сообщения, обычно, являются асинхронными, то есть могут возникать в произвольные моменты времени. Передача такого сообщения, как правило, сопровождается получением фокуса управления принявшим его объектом;

третья разновидность явно обозначает асинхронное сообщение между двумя объектами в некоторой процедурной последовательности. Примером такого сообщения может служить прерывание операции при возникновении исключительной ситуации. В этом случае информация о такой ситуации передается вызывающему объекту для продолжения процесса дальнейшего взаимодействия;

четвертая разновидность сообщения используется для возврата из вызова процедуры. Примером может служить простое сообщение о завершении некоторых вычислений без предоставления результата расчетов объекту-клиенту. В процедурных потоках управления эта стрелка может быть опущена, поскольку ее наличие неявно предполагается в конце активизации объекта. Считается, что каждый вызов процедуры имеет свою пару - возврат вызова. Для непроцедурных потоков управления, включая параллельные и асинхронные сообщения, стрелка возврата должна указываться явным образом.

Предполагается, что время передачи сообщения достаточно мало по сравнению с процессами выполнения действий объектами, то есть, за время передачи сообщения с соответствующими объектами не может произойти никаких изменений. Если же это предположение не может быть признано справедливым, то стрелка сообщения изображается под некоторым наклоном, так чтобы конец стрелки располагался ниже ее начала.

В отдельных случаях объект может посылать сообщения самому себе, инициируя так называемые рефлексивные сообщения. Такие сообщения изображаются прямоугольником со

стрелкой, начало и конец которой совпадают. Подобные ситуации возникают, например, при обработке нажатий на клавиши клавиатуры при вводе текста в редактируемый документ, при наборе цифр номера телефона абонента.

Таким образом, в языке UML каждое сообщение ассоциируется с некоторым действием, которое должно быть выполнено принявшим его объектом. Действие может иметь некоторые аргументы или параметры, в зависимости от конкретных значений которых может быть получен различный результат. Соответствующие параметры будет иметь и вызывающее это действие сообщение. Более того, значения параметров отдельных сообщений могут содержать условные выражения, образуя ветвление или альтернативные пути основного потока управления.

Ветвление потока управления

Для изображения ветвления потока управления рисуются две или более стрелки, выходящие из одной точки фокуса управления объекта. При этом соответствующие условия должны быть явно указаны рядом с каждой из стрелок в форме сторожевого условия. Сторожевые условия должны взаимно исключать одновременную передачу альтернативных сообщений.

Стереотипы сообщений

В языке UML предусмотрены некоторые стандартные действия, выполняемые в ответ на получение соответствующего сообщения. Эти действия могут быть явно указаны на диаграмме последовательности в форме стереотипа рядом с сообщением, к которому относятся. В этом случае они записываются в кавычках. Используются следующие стереотипы сообщений:

«call» (вызвать) - сообщение, требующее вызова операции или процедуры принимающего объекта. Если сообщение с этим стереотипом рефлексивное, то оно инициирует локальный вызов операции у самого пославшего это сообщение объекта;

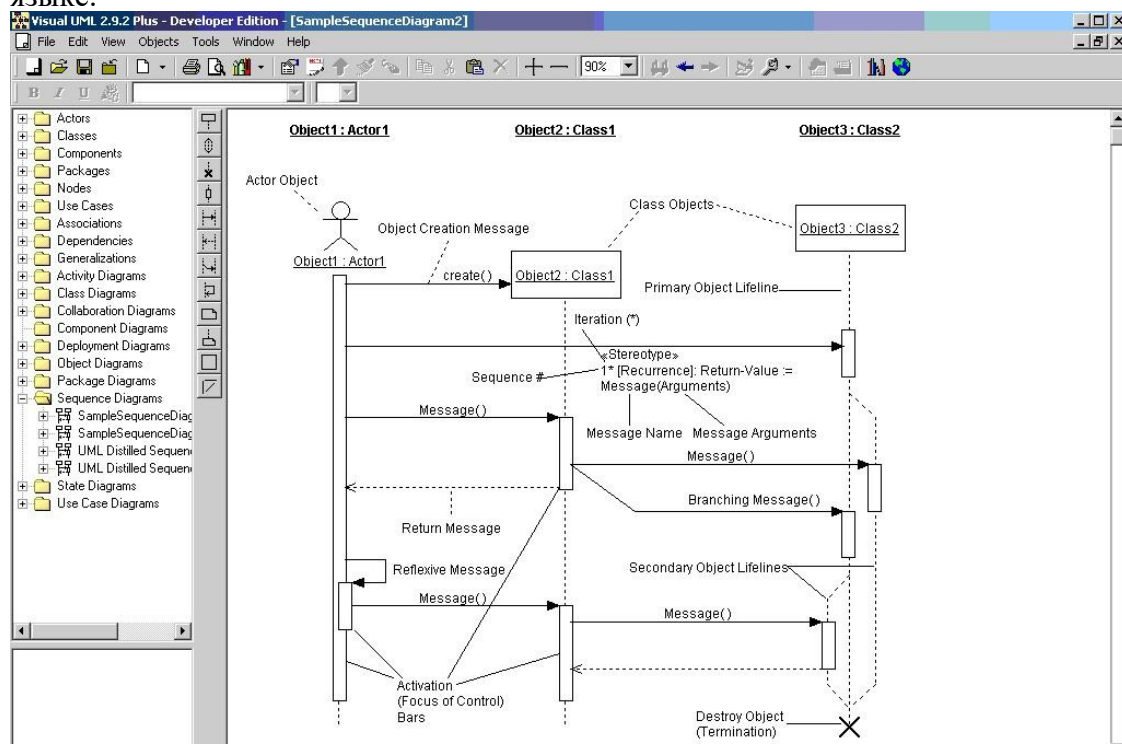
«return» (возвратить) - сообщение, возвращающее значение выполненной операции или процедуры вызвавшему ее объекту. Значение результата может инициировать ветвление потока управления;

«create» (создать) - сообщение, требующее создания другого объекта для выполнения определенных действий. Созданный объект может получить фокус управления, а может и не получить его;

«destroy» (уничтожить) - сообщение с явным требованием уничтожить соответствующий объект. Посылается в том случае, когда необходимо прекратить нежелательные действия со стороны существующего в системе объекта, либо когда объект больше не нужен и должен освободить задействованные им системные ресурсы;

«send» (послать) - обозначает посылку другому объекту некоторого сигнала, который асинхронно инициируется одним объектом и принимается другим. Отличие сигнала от сообщения заключается в том, что сигнал должен быть явно описан в том классе, объект которого инициирует его передачу.

Кроме стереотипов, сообщения могут иметь собственное обозначение операции, вызов которой они инициируют у принимающего объекта. В этом случае рядом со стрелкой записывается имя операции с круглыми скобками, в которых могут указываться параметры или аргументы соответствующей операции. Если параметры отсутствуют, то скобки все равно должны присутствовать после имени операции. Примерами таких операций могут служить следующие: «выдать клиенту наличными сумму (n)», «установить соединение между абонентами (a, b)», «сделать вводимый текст невидимым ()», «подать звуковой сигнал тревоги ()».



Построение диаграммы состояний

Настройка

1. В меню модели выберите пункт Tools > Options (Инструменты > Параметры).
2. Перейдите на вкладку диаграмм.
3. Контрольные переключатели Sequence Numbering, Collaboration Numbering и Focus of Control должны быть помечены.
4. Нажмите ОК, чтобы выйти из окна параметров.

Создание диаграммы Последовательности

1. Щелкните правой кнопкой мыши на Логическом представлении браузера.
2. В открывшемся меню выберите пункт New > Sequence Diagram.
3. Назовите новую диаграмму.
4. Дважды щелкните на ней, чтобы открыть ее.

Добавление на диаграмму действующего лица и объектов

1. Перетащите действующее лицо с браузера на диаграмму.
 2. На панели инструментов нажмите кнопку Object (Объект).
 3. Щелкните мышью в верхней части диаграммы, чтобы поместить туда новый объект.
 4. Назовите объект.
5. Повторите этапы 3 и 4, чтобы поместить на диаграмму все остальные объекты.

Добавление сообщений на диаграмму

1. На панели инструментов нажмите кнопку Object Message (Сообщение объекта).
 2. Проведите мышью от линии жизни одного объекта к линии жизни другого объекта.
- Выделив сообщение, введите его имя

Порядок выполнения

1. Постройте диаграммы состояний и последовательностей или, если требуется, несколько диаграмм.
 2. Ответьте на контрольные вопросы.

Контрольные вопросы

1. Что такое объект в объектном моделировании?
2. Какие типы отношений имеются в нотации UML?
3. Назовите основные стереотипы класса.
4. Назовите основные элементы диаграммы состояний
5. Назовите основные элементы диаграммы последовательности.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №21

Тема: Диаграммы состояний и последовательности

Цель: изучение объектно-ориентированного моделирования и исследование процесса построения диаграммы состояний и последовательности в заданной предметной области с помощью пакета Rational Rose 2000.

Диаграмма состояний (statechart diagram)

Главное предназначение этой диаграммы — описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели

в течение его жизненного цикла. Диаграмма состояний представляет динамическое поведение сущностей, на основе спецификации их реакции на восприятие некоторых конкретных событий.

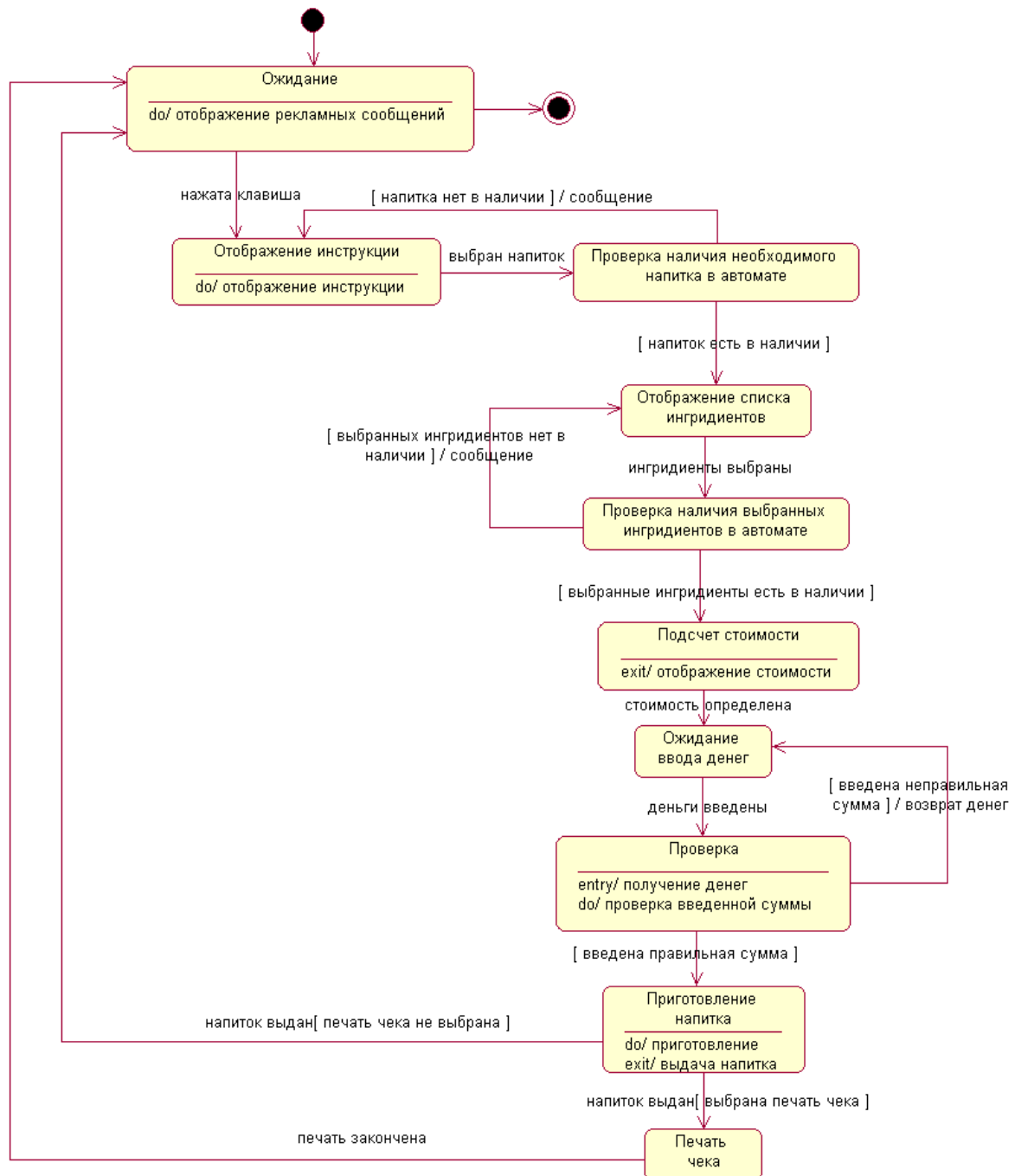


Диаграмма последовательности (sequence diagram)

Для моделирования взаимодействия объектов в языке UML используются соответствующие диаграммы взаимодействия. Взаимодействия объектов можно рассматривать во времени, и тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности. Взаимодействующие объекты обмениваются между собой некоторой информацией. При этом информация принимает

форму законченных сообщений. Другими словами, хотя сообщение и имеет информационное содержание, оно приобретает дополнительное свойство оказывать направленное влияние на своего получателя.

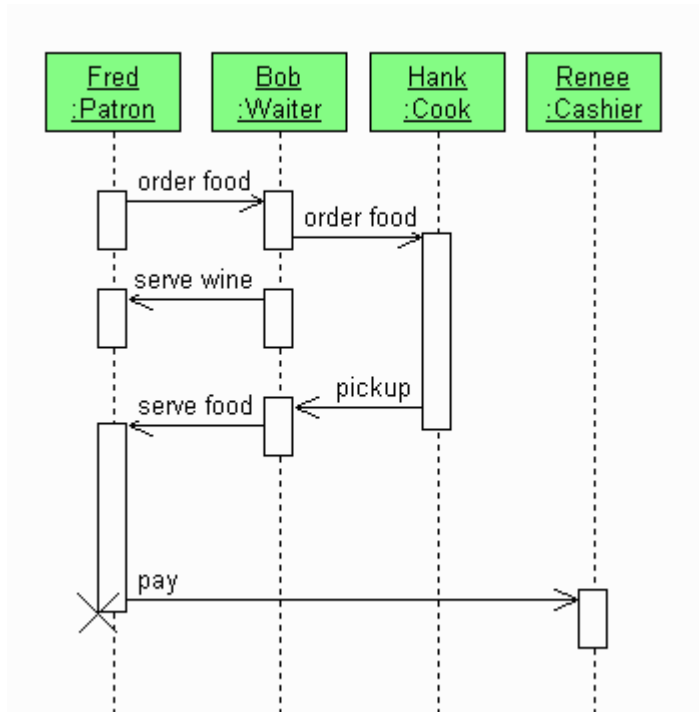


Диаграмма кооперации (collaboration diagram)

На диаграмме кооперации в виде прямоугольников изображаются участвующие во взаимодействии объекты, содержащие имя объекта, его класс и, возможно, значения атрибутов. Как и на диаграмме классов, указываются ассоциации между объектами в виде различных соединительных линий. При этом можно явно указать имена ассоциации и ролей, которые играют объекты в данной ассоциации. В отличие от диаграммы последовательности, на диаграмме кооперации изображаются только отношения между объектами, играющими определенные роли во взаимодействии.

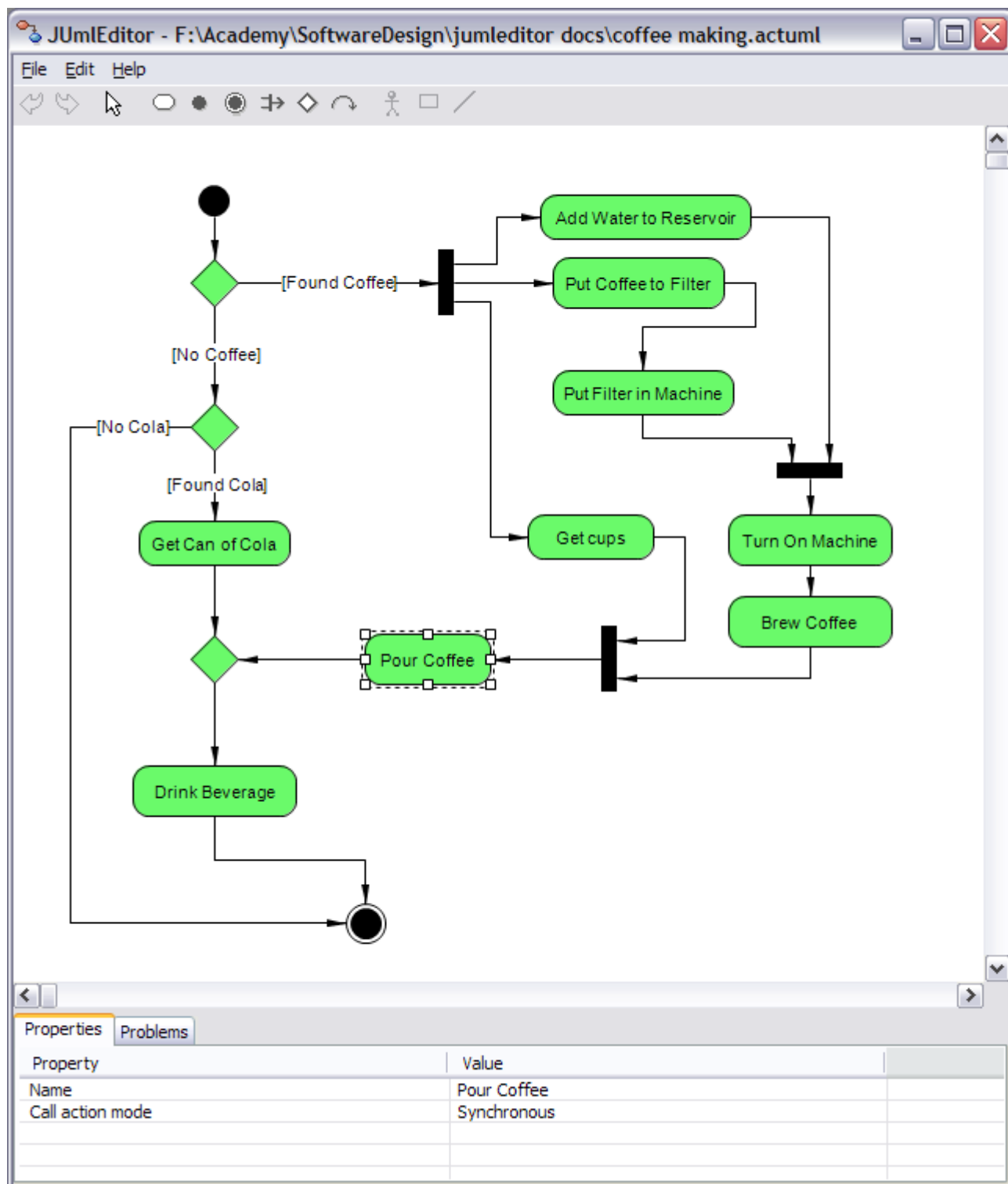


Диаграмма компонентов (component diagram)

Диаграмма компонентов, в отличие от ранее рассмотренных диаграмм, описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Во многих средах разработки модуль или компонент соответствует файлу.

Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции исходных текстов программ. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

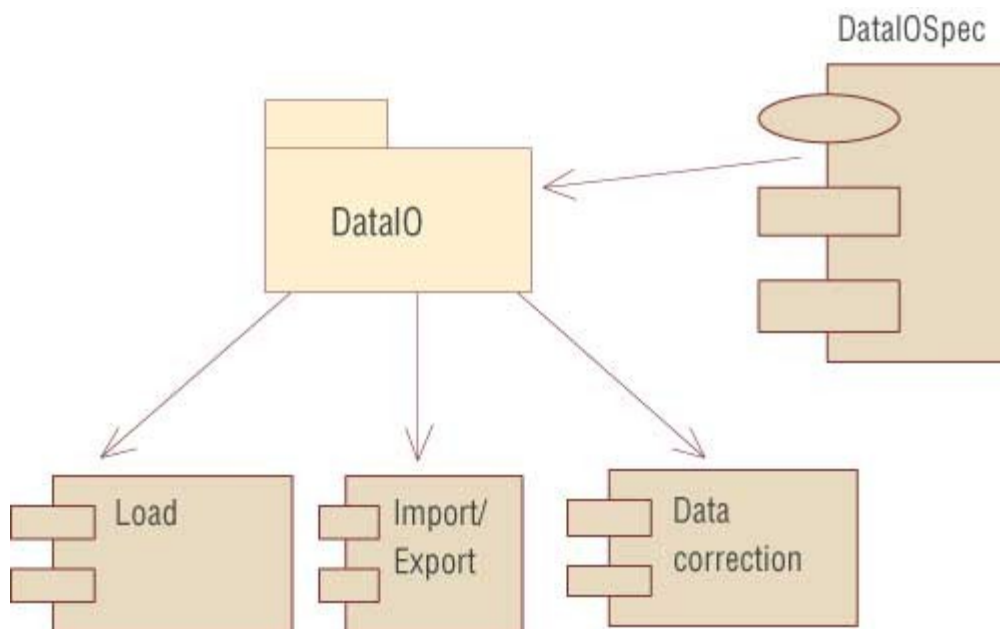
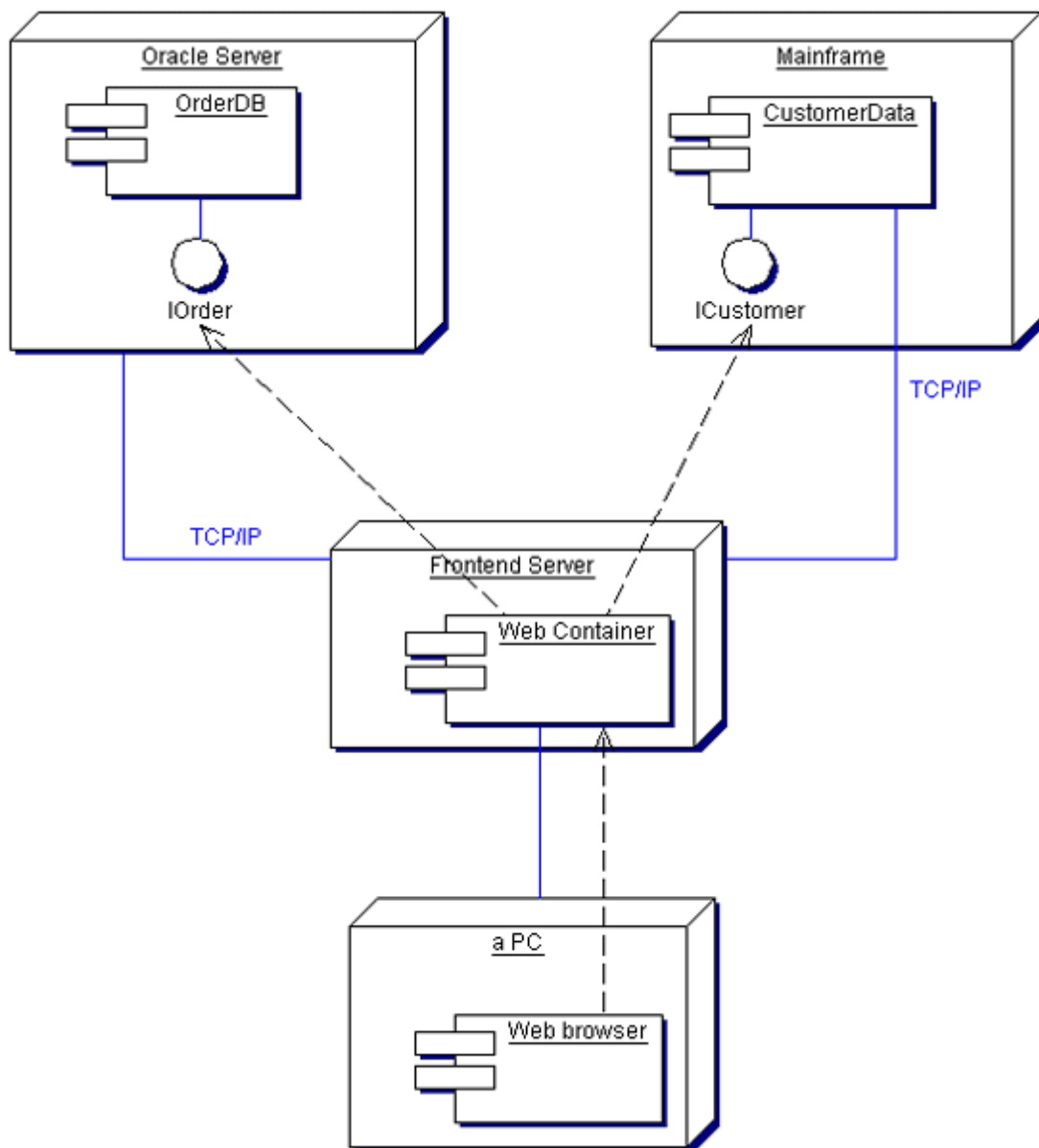


Диаграмма развертывания (deployment diagram)

Диаграмма развертывания предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения (runtime). При этом представляются только компоненты-экземпляры программы, являющиеся исполнимыми файлами или динамическими библиотеками. Те компоненты, которые не используются на этапе исполнения, на диаграмме развертывания не показываются. Диаграмма развертывания содержит графические изображения процессоров, устройств, процессов и связей между ними. В отличие от диаграмм логического представления, диаграмма развертывания является единой для системы в целом, поскольку должна всецело отражать особенности ее реализации. Эта диаграмма, по сути, завершает процесс ООАП для конкретной программной системы и ее разработка, как правило, является последним этапом спецификации модели.



На этом закончим обзорный экскурс по диаграммам в частности и проектированию в общем. Стоит отметить, что процесс проектирования уже давно стал стандартом разработки ПО, но часто приходится сталкиваться с великолепно написанной программой, которая из-за отсутствия нормальной документации обрастает ненужным побочным функционалом, костылями, становится громоздкой и теряет былое качество.

Основная литература:

1. Грекул В.И., Коровкина Н.Л., Куприянов Ю.В. Проектирование информационных систем. М.: Национальный открытый университет «ИНТУИТ», 2016, Режим доступа: <http://www.iprbookshop.ru/67376.html>
2. Вичугова А.А. Инструментальные средства разработки компьютерных систем и комплексов [Электронный ресурс] : учебное пособие для СПО / А.А. Вичугова. — Электрон. текстовые данные. — Саратов: Профобразование, 2017. — 135 с. — 978-5-4488-0015-3. — Режим доступа: <http://www.iprbookshop.ru/66387.html>
3. Долженко А.И. Технологии командной разработки программного обеспечения информационных систем [Электронный ресурс]/ Долженко А.И.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 300 с.— Режим доступа: <http://www.iprbookshop.ru/39569.html>.— ЭБС «IPRbooks»

Дополнительная литература:

1. Рудаков А. Технология разработки программных продуктов: учебник. Изд.Academia. Среднее профессиональное образование. 2013 г. 208 стр.