

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

# **ОРГАНИЗАЦИЯ И ПЛАНИРОВАНИЕ РАЗРАБОТКИ ИНФОРМАЦИОННЫХ СИСТЕМ**

Методические указания по выполнению практических занятий  
для студентов направления подготовки 09.03.02 –  
Информационные системы и технологии

Невинномысск 2021

Настоящие методические указания предназначены для студентов направления подготовки 09.03.02 – Информационные системы и технологии. Они разработаны в соответствии с федеральным государственным образовательным стандартом и основной образовательной программой направления.

В методических указаниях приведены основные сведения о языке визуального моделирования UML, В методических указаниях приведены основные сведения об организации лабораторного практикума, предложена примерная структура лабораторных работ, кратко изложено содержание их разделов и приведен список рекомендуемых литературных источников.

Составитель      *канд. техн. наук, доцент Д.В. Болдырев*

Отв. редактор    *канд. техн. наук А.А. Евдокимов*

## Тематический план практических работ

№ Темы дисциплины	Наименование тем дисциплины, их краткое содержание	Объем часов	Из них в интерактивной форме
4 семестр			
<b>Тема 2. Каноническое проектирование</b>			
1	Варианты использования и действующие лица	1.50	Решение типовых задач
2	Взаимодействие объектов	1.50	Решение типовых задач
3	Классы и пакеты	1.50	Решение типовых задач
4	Атрибуты и операции	1.50	Решение типовых задач
5	Связи	1.50	Решение типовых задач
6	Поведение объектов	1.50	Решение типовых задач
7	Представление компонентов	1.50	Решение типовых задач
8	Представление размещения	1.50	Решение типовых задач
	<b>Итого за 3 семестр</b>	<b>12,00</b>	
	<b>Итого</b>	<b>12,00</b>	

\* - с применением дистанционных образовательных технологий

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ .....	5
ОБЩИЕ СВЕДЕНИЯ.....	7
Практическое занятие № 1 Варианты использования и действующие лица .....	13
Практическое занятие № 2 Взаимодействие объектов.....	17
Практическое занятие № 3 Классы и пакеты .....	35
Практическое занятие № 4 Атрибуты и операции.....	43
Практическое занятие № 5 Связи .....	48
Практическое занятие № 6 Поведение объектов .....	51
Практическое занятие № 7 Представление компонентов .....	56
Практическое занятие № 8 Представление размещения .....	63
СПИСОК ЛИТЕРАТУРЫ .....	66
ПРИЛОЖЕНИЕ.....	67

## ВВЕДЕНИЕ

Изучение дисциплины «Проектирование единого информационного пространства виртуальных предприятий» имеет целью систематизацию, закрепление и расширение теоретических знаний по формированию информационной базы предприятий и организаций, развитие у студентов навыков самостоятельного анализа предметной области и разработки проектных решений.

Согласно ФГОС ВО направления 09.03.02 – Информационные системы и технологии в результате освоения дисциплины студент должен знать:

- основные положения о едином информационном пространстве виртуальных предприятий, виды обеспечения и программно-технические средства для построения интегрированных систем проектирования и управления, информационно-функциональную интеграцию автоматизированных систем различного назначения;
- инструментальные средства проектирования систем управления, стандарты и языки представления информационных моделей продукции;
- основные принципы проектирования систем автоматизации и управления объектами различного служебного назначения в режиме реального времени;
- основные понятия об интегрированных системах проектирования и управления автоматизированных и автоматических производств различного назначения, их функции и структуру.

В результате освоения дисциплины студент должен уметь:

- разрабатывать техническое задание и техническое предложение на разработку автоматизированных систем;
- применять методику объективно-ориентированного подхода при проектировании систем автоматизации и управления.

В результате освоения дисциплины студент должен владеть:

- навыками построения виртуальных предприятий, их элементов, использования стандартов и языков моделей продукции;
- навыками разработки бизнес-планов и оценки экономической эффективности проводимых мероприятий в области автоматизации технологических процессов и производств, управления жизненным циклом продукции и ее качеством;
- навыками разработки распределенных компьютерных информационно-управляющих систем, информационных моделей знаний.

Лабораторный практикум имеет целью выработку у студентов навыков в трех направлениях:

1. Применение соответствующих методологий для разработки информационных систем и программного обеспечения;
2. Применение языка UML для моделирования и проектирования информационных систем;
3. Применение соответствующего программного инструментария.

## ОБЩИЕ СВЕДЕНИЯ

### 1. Введение в Rational Rose

Компания Rational Software является лидирующей в области создания методологий и программных решений, ориентированных на программистов, аналитиков, тестировщиков. Спектр выпускаемого программного обеспечения целиком покрывает потребность всех участников проекта: от аналитиков до разработчиков и внедренцев. Все программно-методологические решения — плод многолетнего труда аналитиков и разработчиков как самой Rational, так и ее партнеров. В итоге все решения были собраны воедино. Так появился RUP — Rational Unified Process — методологическая энциклопедия, в которой описаны все шаги, необходимые для создания качественного программного продукта. Пользуясь подобной энциклопедией и применяя соответствующие инструменты, рекомендуемые Rational, команда будет создавать обеспечение качественно и в срок. «Строй быстрее и качественней!» — вот лозунг, выдвигаемый Rational.

Особое место в RUP занимают проектирование и конфигурационное управление. Особо выделяются они потому, что те два инструмента, которые поддерживаются на данных этапах (Rational Rose и Rational ClearCase), используются на протяжении всего жизненного цикла разработки программного обеспечения. Если Rose со своей стороны используют и проектировщики, и разработчики, и аналитики (практически добрая половина коллектива), то ClearCase используют все, поскольку результатом любой деятельности в мире информационных технологий является файл, который где-то необходимо хранить и не просто хранить, а знать все изменения, которые были в него внесены на каждом этапе разработки информационной системы.

Rational Rose, являясь объектно-ориентированным средством проектирования, способна моделировать ситуации любой сложно-

сти: от проектирования банковской системы до разработки кода на C++. В умелых руках Rose неоценимый инструмент!

Rational Rose в отличие от подобных средств проектирования способна проектировать системы любой сложности, то есть инструментарий программы допускает как высокоуровневое (абстрактное) представление (например, схема автоматизации предприятия), так и низкоуровневое проектирование (интерфейс программы, схема базы данных, частичное описание классов). Вся мощь программы базируется всего на 7 диаграммах, которые в зависимости от ситуации способны описывать различные действия.

Давайте попробуем разобраться, что же реально даст данный инструмент отдельным участникам проекта: проектировщикам, аналитикам, разработчикам.

*Проектировщикам.* В большинстве случаев подрядчик не может написать качественное программное обеспечение в установленный срок, поскольку заранее не были оговорены с заказчиком наиболее важные моменты в системе. Например, заказчик забыл рассказать об очень важной составляющей своей системы, а вспомнил об этом, когда увидел (запустил) предложенную готовую программу. Соответственно, проблема на данном этапе — взаимопонимание двух сторон. Значит, если проектировщик с заказчиком посредством моделирования предметной области в Rational Rose четко и скрупулезно описали каждую деталь и увидели ее в виде понятных диаграмм, то проблему взаимопонимания можно отбросить. Таким образом, при разработке современных информационных систем много времени уделяется проектированию (моделированию предметной области), поскольку необходимо изначально договориться с заказчиком обо всех нюансах, а не вносить в режиме «пожарной команды» изменения на более поздних этапах. То есть Rational Rose поможет на концептуальном уровне разобраться с генеральным планом автоматизации. Для улучшения взаимопонимания обеих сторон совместно с Rose применяют инструмент SoDA, позволяющий на основе построенной модели дать полный отчет по ее состоянию, соответствующий всем общепризнанным мировым



стандартам (в частности ISO 9000). Как видим, внедрение Rose на предприятии позволяет, в дополнение к вышеописанному, структурировать сопроводительную документацию, привести ее к необходимому стандарту с минимальными девиациями.

*Разработчикам.* Не меньше возможностей Rose дает и разработчикам. Давайте снова повторим очевидную вещь: информационные системы конца 90 гг. вышли на такой уровень сложности, что справиться с ними уже под силу только крупным компаниям с большим количеством узкоспециализированных разработчиков. Времена программистов-одиночек ушли в небытие. В современных условиях механизм «сам все сделаю» дает явный сбой. В качестве второй проблемы можно отметить некоторую текучесть кадров на отдельно взятом предприятии. Каждый раз, при включении нового сотрудника в проект, необходимо посвящать его во все детали проекта, на что уходит драгоценное время коллег, отрываемых от основной работы. При наличии же Rose достаточно показать все диаграммы проекта и предоставить проектную документацию, сгенерированную на основе полученной модели, как все станет на свои места. Разработчик увидит как весь проект в целом, так и свою часть. Конкретно же в плане разработки Rose поддерживает проектирование, основанное на двух способах: прямом и обратном. В режиме прямого проектирования разработчик рисует определенным образом диаграммы классов и их взаимодействия, а на выходе получает сгенерированный код на определенном языке программирования. В режиме же обратного проектирования возможно построение модели на базе имеющегося исходного кода. Из этого следует самая главная возможность для разработчиков: повторное проектирование (Round-trip), когда разработчик описывает классы в Rose, генерирует код, дописывает необходимую полнофункциональную часть и снова закидывает в Rose, для представления того, что же система получила в результате его действий.

Важнейшим свойством Rational Rose принято считать открытость архитектуры, что позволяет дополнять имеющийся в ней инструментарий новыми функциями. Например, в стандартном вари-

анте Rose не поддерживает кодогенерацию на Ассемблере: Путем написания дополнительного модуля подобную проблему можно решить. Спешу обрадовать, на Западе достаточно компаний, выпускающих подобные модули расширения для различных языков программирования, правда, на Ассемблере еще нет, но мы будем надеяться!

Вот список включенных стандартных модулей: C++, ADA, CORBA, Visual Basic, XML, COM, Oracle). То есть Rational Rose способна проводить прямое и обратное проектирование в данных системах.

Подведем первые итоги того, что может делать Rational Rose

- Проектировать системы любой сложности
- Давать развернутое представление о проекте в сочетании со средствами документирования (SoDA)
  - Проводить кодогенерацию
  - Проводить обратное проектирование имеющихся систем
  - Имеет открытый для дополнений интерфейс
  - Интегрируется со средствами разработки (Visual Studio)
  - Поддержка языка UML
  - Наличие средств автоматического контроля, в том числе проверки соответствия двух моделей
    - Удобный для пользователя графический интерфейс
    - Многоплатформенность
    - Интегрируемость с другими инструментальными средствами, поддерживающими жизненный цикл программных систем, в том числе со средством управления требованиями (Requisite Pro), со средствами тестирования (SQA Suite, Performance Studio), со средствами конфигурационного управления (ClearCase, PVCS).

Являясь объектно-ориентированным инструментом моделирования, Rose базируется на UML (Universal Modeling Language) — универсальном языке моделирования, который был разработан компанией Rational именно с целью создания наиболее оптимального и универсального языка для описания как предметной обла-

сти, так и конкретной задачи в программировании. Любая задача программируется при помощи определенных диаграмм. UML поддерживает построение следующих диаграмм:

- Activity diagram (диаграммы описаний технологий, процессов, функций).
- Use case diagram (диаграммы функций).
- Class diagram (диаграммы классов).
- State diagram (диаграммы состояний);
- Sequence diagram (диаграммы последовательностей действий);
- Collaboration diagram (диаграммы взаимодействий)
- Component diagram (диаграммы компонент);
- Deployment diagram (диаграммы топологии).

Соответственно, Rational Rose является инструментом, который позволяет строить указанные диаграммы при проектировании программных систем. К сожалению, объем статьи не позволяет описать назначение всех диаграмм и спецификаций! Но мы попробуем разобраться в инструменте с точки зрения разработчика, для простоты используя только один тип диаграмм — Class Diagramm.

Все разработчики сталкиваются с ситуацией, когда приходится проектировать большие классы. При ручном вводе и объявлении имеется ряд подводных камней: во-первых, постановщик задач, как правило, описывает «что нужно» на словах, в крайнем случае, с минимальным бумажным сопровождением; во-вторых, разработчик, создающий систему, опять-таки в большинстве случаев игнорирует все комментарии, которыми необходимо сопровождать программный код. Что же получается в итоге? Постановщик задач путается в программе, разработчик сам не помнит, что к чему, а если на его место взят новый сотрудник: Тут на ум приходит еще одно традиционное для России высказывание разработчика: «мне проще все написать заново». И ведь пишут: Тормозя производство программного продукта. Дело в том, что к разработке ПО относятся как к искусству, а необходимо относиться, как к производственному процессу со строгим распределением ролей, полномочий и пр.

## **2. Порядок выполнения работ**

Варианты индивидуального задания определяются преподавателем.

Для выполнения всех лабораторных работ предлагается единый порядок, предусматривающий следующие шаги

1. Ознакомиться с постановкой задачи и исходными данными.
2. Разработать предлагаемую в работе диаграмму
3. Реализовать разработанную диаграмму в среде Rational Rose
4. Сохранить файл модели
5. Составить отчет по проделанной работе

## **3. Содержание отчета**

Отчет оформляется по каждой лабораторной работе и состоит из следующих разделов:

1. Тема лабораторной работы.
2. Цель работы.
3. Индивидуальное задание.
4. Описание необходимых абстракций
5. Разработанная диаграмма

## **Практическое занятие № 1**

### **Варианты использования и действующие лица**

#### **Создание диаграммы Вариантов Использования**

Создайте диаграмму Вариантов Использования для своего индивидуального задания.

#### **Индивидуальное задание:**

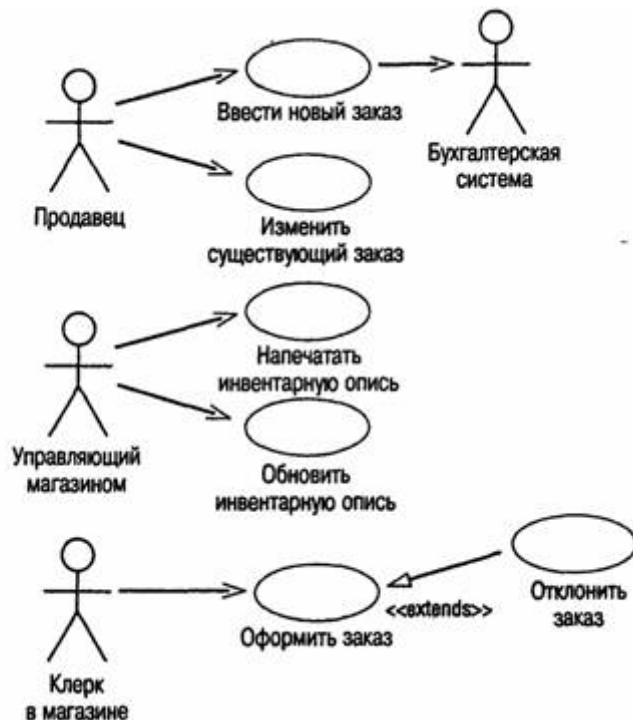
1. Создать диаграмму Вариантов Использования, задать варианты использования и действующих лиц
2. Создать абстрактный вариант использования
3. Добавить ассоциации
4. Добавить связи расширения
5. Добавить описания к вариантам использования
6. Добавить описания к действующим лицам
7. Прикрепить файл к варианту использования
8. Сохранить файл модели, составить отчет

#### **Пример выполнения упражнения**

#### **Создание диаграммы Вариантов Использования, вариантов использования и действующих лиц**

1. Дважды щелкнув мышью на Главной диаграмме Вариантов Использования (Main) в браузере, откройте ее.
2. С помощью кнопки Use Case (Вариант использования) панели инструментов поместите на диаграмму новый вариант использования.
3. Назовите его «Ввести новый заказ».
4. Повторив этапы 2 и 3, поместите на диаграмму остальные варианты использования: Изменить существующий заказ, Напечатать инвентарную опись, Обновить инвентарную опись, Оформить заказ, Отклонить заказ.
5. С помощью кнопки Actor (Действующее лицо) панели инструментов поместите на диаграмму новое действующее лицо.
6. Назовите его «Продавец».

7. Повторив шаги 5 и 6, поместите на диаграмму остальных действующих лиц: Управляющий магазином, Клерк магазина, Бухгалтерская система.



Пример диаграммы вариантов использования

### Создание абстрактного варианта использования

1. Щелкните правой кнопкой мыши на варианте использования «Отклонить заказ» на диаграмме.
2. В открывшемся меню выберите пункт Open Specification (Открыть спецификацию).
3. Установите флажок Abstract (Абстрактный), чтобы сделать этот вариант использования абстрактным.

### Добавление ассоциаций

1. С помощью кнопки Unidirectional Association (Однонаправленная ассоциация) панели инструментов нарисуйте ассоциацию между действующим лицом Продавец и вариантом использования «Ввести новый заказ».
2. Повторив шаг 1, поместите на диаграмму остальные ассоциации.

### **Добавление связи расширения**

1. С помощью кнопки Generalization (Обобщение) панели инструментов нарисуйте связь между вариантом! использования «Отклонить заказ» и вариантом использования «Оформить заказ». Стрелка должна идти от первого варианта использования ко второму. Связь расширения означает, что вариант использования «Отклонить заказ» при необходимости дополняет функциональные возможности варианта использования «Оформить заказ».

2. Щелкните правой кнопкой мыши на новой связи между вариантами использования «Отклонить заказ» и «Оформить заказ».

3. В открывшемся меню выберите пункт Open Specification (Открыть спецификацию).

4. В раскрывающемся списке стереотипов введите слово extends (расширение), затем нажмите ОК.

5. Надпись «extends» появится на линии данной связи.

### **Добавление описаний к вариантам использования**

1. Выделите в браузере вариант использования «Ввести новый заказ».

2. В окне документации введите следующее описание: «Этот вариант использования дает клиенту возможность ввести новый заказ в систему».

3. С помощью окна документации добавьте описания ко всем остальным вариантам использования.

### **Добавление описаний к действующему лицу**

1. Выделите в браузере действующее лицо Продавец.

2. В окне документации введите следующее описание: «Продавец — это служащий, старающийся продать товар».

3. С помощью окна документации добавьте описания к остальным действующим лицам.

### **Прикрепление файла к варианту использования**

1. Создайте свой собственный файл OrderFlow.doc и внесите в него поток событий, как показано ниже. Основной поток событий для варианта использования «Ввести новый заказ»

- *Продавец выбирает в имеющемся меню пункт “Создать новый заказ”.*
- *Система выводит форму “Детали заказа”.*
- *Продавец вводит номер заказа, заказчика и то, что заказано.*
- *Продавец сохраняет заказ.*
- *Система создаёт новый заказ и сохраняет его в базе данных.*

1. Щелкните правой кнопкой мыши на варианте использования «Ввести новый заказ».

2. В открывшемся меню выберите пункт Open Specification (Открыть спецификацию).

3. Перейдите на вкладку Files (Файлы).

4. Щелкните правой кнопкой мыши в белой области и в открывшемся меню выберите пункт Insert File (Вставить файл).

5. Укажите файл OpenFlow.doc и нажмите на кнопку Open (Открыть), чтобы прикрепить файл к варианту использования.



## Практическое занятие № 2

### Взаимодействие объектов

Создайте диаграммы Последовательностей и Кооперации для своего индивидуального задания.

#### **Индивидуальное задание:**

1. Создать диаграмму Последовательностей
2. Создать диаграмму Кооперации
3. Сохранить файл модели, составить отчет

Пример выполнения упражнения описан ниже.

#### **Пример выполнения упражнения**

##### Постановка задачи

Поговорив с Бобом, Сюзан поняла, что должна делать система обработки заказов, создаваемая ею для фирмы Robertson's Cabinets. Она нарисовала диаграмму Вариантов Использования. Изучив эту диаграмму, все пришли к согласию по поводу области применения системы.

Теперь наступило время анализа ее составных частей. Высший приоритет среди пользователей имеет вариант использования «Ввести новый заказ», он же связан с наибольшим риском. Сюзан решила заняться им в первую очередь.

Она поговорила с Карлом, заведующим отделом продаж. Вдвоем они обсудили поток событий, который будет реализовываться в варианте использования.

Получив нужную информацию, Сюзан составила описание сценариев:

- Продавец вводит новый заказ.
- Продавец пытается ввести заказ, но товара нет на складе.
- Продавец пытается ввести заказ, но при его сохранении в базе данных возникает ошибка.

Затем она приступила к созданию диаграмм Последовательности и Кооперативных диаграмм для сценария «Ввести новый заказ».

### Создание диаграмм Взаимодействия

Создайте диаграмму Последовательности и Кооперативную диаграмму, отражающую ввод нового заказа в систему обработки заказов. Готовая диаграмма Последовательности показана на рис. 1.

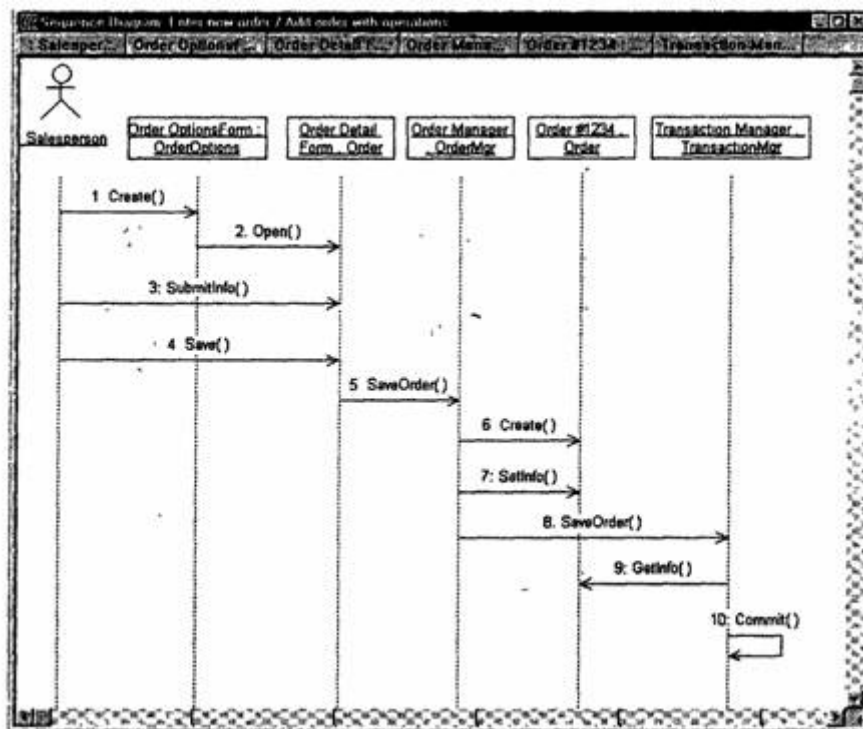


Рис. 1. Диаграмма Последовательности для ввода нового заказа

Это только одна из диаграмм, необходимых для моделирования варианта использования «Ввести новый заказ». Она соответствует успешному варианту хода событий. Для описания того, что случится, если возникнет ошибка или если пользователь выберет другие действия из предложенных, придется разработать дополнительные диаграммы. Каждый «альтернативный поток варианта использования может быть промоделирован с помощью собственных диаграмм Взаимодействия».

## **Этапы выполнения упражнения**

### **Настройка**

1. В меню модели выберите пункт Toots > Options (Инструменты > Параметры).
2. Перейдите на вкладку Diagram (Диаграмма).
3. Установите флажки Sequence numbering, Collaboration numbering и Focus of control.
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 (Создать > Диаграмма Последовательности).
3. Назовите новую диаграмму Add order (Ввод заказа).
4. Дважды щелкнув на этой диаграмме, откройте ее.

### **Добавление на диаграмму действующего лица и объектов**

1. Перетащите действующее лицо Salesperson (Продавец) из браузера на диаграмму.
2. Нажмите кнопку Object (Объект) панели инструментов.
3. Щелкните мышью в верхней части диаграммы, чтобы поместить туда новый объект.
4. Назовите объект Order Options Form (Выбор варианта заказа).
5. Повторив шаги 3 и 4, поместите на диаграмму объекты:
  - Order Detail Form (Форма деталей заказа)
  - Order N1234 (Заказ №1234)

### **Добавление сообщений на диаграмму**

1. На панели инструментов нажмите кнопку Object Message (Сообщение объекта).

2. Проведите мышью от линии жизни действующего лица Salesperson (Продавец) к линии жизни объекта Order Options Form (Выбор варианта заказа).

3. Выделив сообщение, введите его имя — Create new order (Создать новый заказ).

4. Повторив шаги 2 и 3, поместите на диаграмму сообщения:

- Open form (Открыть форму) — между Order Options Form и Order Detail Form

- Enter order number, customer, order items (Ввести номер заказа, заказчика и число заказываемых предметов) — между Salesperson и Order Detail Form

- Save the order (Сохранить заказ) — между Salesperson и Order Detail Form

- Create new, blank order (Создать пустой заказ) — между Order Detail Form и Order N1234

- Set the order number, customer, order items (Ввести номер заказа, заказчика и число заказываемых предметов) — между Order Detail Form и Order N1234

- Save the order (Сохранить заказ) — между Order Detail Form и Order N1234

Завершен первый этап работы. Готовая диаграмма Последовательности представлена на рис. 2. Теперь нужно позаботиться об управляющих объектах и о взаимодействии с базой данных. Как видно из диаграммы, объект Order Detail Form имеет множество ответственностей, с которыми лучше всего мог бы справиться управляющий объект. Кроме того, новый заказ должен сохранять себя в базе данных сам. Вероятно, эту обязанность лучше было бы переложить на другой объект.

### **Добавление на диаграмму дополнительных объектов**

1. Нажмите кнопку Object панели инструментов.

2. Щелкните мышью между объектами Order Detail Form и Order N1234, чтобы поместить туда новый объект.

3. Введите имя объекта — Order Manager (Управляющий заказами).
4. Нажмите кнопку Object панели инструментов.
5. Новый объект расположите справа от Order N1234.
6. Введите его имя — Transaction Manager (Управляющий транзакциями).

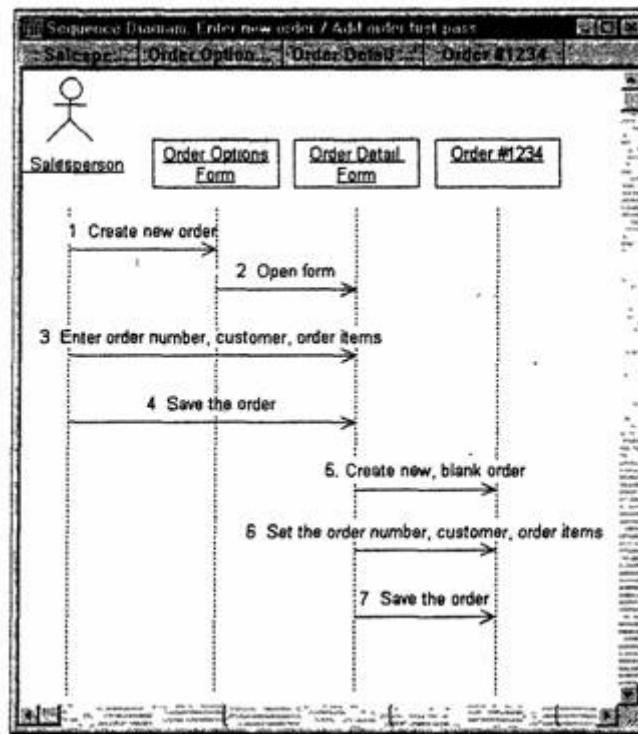


Рис. 2 — Диаграмма Последовательности ввода нового заказа после завершения первого этапа работы

### Назначение ответственностей объектам

1. Выделите сообщение 5: Create new, blank order (Создать пустой заказ).
2. Нажав комбинацию клавиш CTRL+D, удалите это сообщение.
3. Повторите шаги 1 и 2 для удаления двух последних сообщений:
  - Set the order number, customer, order items (Вести номер заказа, заказчика и число заказываемых предметов)
  - Save the order (Сохранить заказ)
4. . Нажмите кнопку Object Message панели инструментов.

5. Поместите на диаграмму новое сообщение, расположив его под сообщением 4 между Order Detail Form и Order Manager.

6. Назовите его Save the order (Сохранить заказ).

7. Повторите шаги 4 — 6, добавив сообщения с шестого по девятое и назвав их:

- Create new, blank order (Создать новый заказ) — между Order Manager и Order N1234

- Set the order number, customer, order items (Вести номер заказа, заказчика и число заказываемых предметов) — между Order Manager и Order N1234

- Save the order (Сохранить заказ) — между Order Manager и Transaction Manager

- Collect order information (Информация о заказе) — между Transaction Manager и Order N1234

8. На панели инструментов нажмите кнопку Message to Self (Сообщение себе).

9. Щелкните на линии жизни объекта Transaction Manager (Управляющий транзакциями) ниже сообщения 9, добавив туда рефлексивное сообщение.

10. Назовите его Save the order information to the database (Сохранить информацию о заказе в базе данных).

Теперь диаграмма Последовательности должна выглядеть, как показано на рис. 3.

### **Соотнесение объектов с классами**

1. Щелкните правой кнопкой мыши на объекте Order Options Form (Выбор варианта заказа).

2. В открывшемся меню выберите пункт Open Specification (Открыть спецификацию).

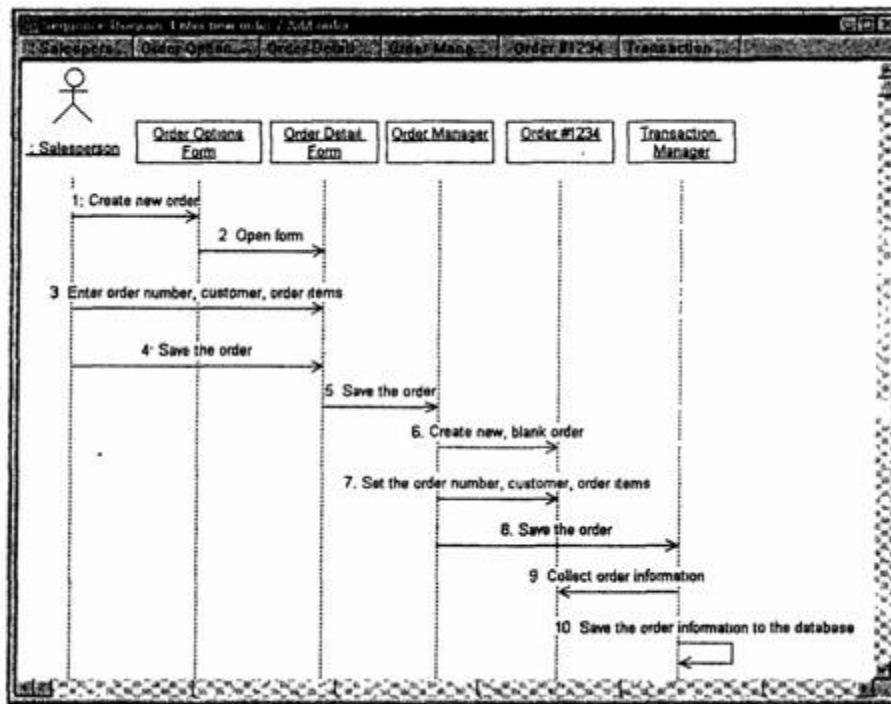


Рис. 3 — Диаграмма Последовательности с новыми объектами

3. В раскрывающемся списке классов выберите пункт (Создать). Появится окно спецификации классов.

4. В поле Name введите OrderOptions (Выбор заказа).

5. Щелкните на кнопке ОК. Вы вернетесь в окно спецификации объекта.

6. В списке классов выберите класс OrderOptions.

7. Щелкните на кнопке ОК, чтобы вернуться к диаграмме. Теперь объект называется Order Options Form : OrderOptions.

8. Для соотнесения остальных объектов с классами повторите шаги с 1 по 7:

- Класс OrderDetail соотнесите с объектом Order Detail Form

- Класс OrderMgr — с объектом Order Manager

- Класс Order — с объектом Order N1234

- Класс TransactionMgr — с объектом Transaction Manager

По завершении этих действий ваша диаграмма должна иметь вид, представленный на рис. 4.

### Соотнесение сообщений с операциями

1. Щелкните правой кнопкой мыши на сообщении 1: Create new order (Создать новый заказ).
2. В открывшемся меню выберите пункт (создать операцию). Появится окно спецификации операции.
3. В поле Name введите имя операции — Create (Создать).
4. Нажмите на кнопку ОК, чтобы закрыть окно спецификации операции и вернуться к диаграмме.
5. Еще раз щелкните правой кнопкой мыши на сообщении 1.
6. В открывшемся меню выберите новую операцию CreateO.
7. Повторите шаги с 1 по 6, чтобы соотнести с операциями все остальные сообщения:
  - Сообщение 2: Open form (Открыть форму) соотнесите с операцией Open()
  - Сообщение 3: Enter order number, customer, order items (Ввести номер заказа, заказчика и число заказываемых предметов) — с операцией SubmitInfo()
  - Сообщение 4: Save the order (Сохранить заказ) — с операцией Save()
  - Сообщение 5: Save the order (Сохранить заказ) — с операцией SaveOrderQ
  - Сообщение 6: Create new, blank order (Создать пустой заказ) — с операцией Create()
  - Сообщение 7: Set the order number, customer, order items (Ввести номер заказа, заказчика и число заказываемых предметов) — с операцией Settrfo()
  - Сообщение 8: Save the order (Сохранить заказ) — с операцией SaveOrderQ
  - Сообщение 9: Collect order information (Информация о заказе) — с операцией GetInfo();
  - Сообщение 10: Save the order information to the database (Сохранить информацию о заказе в базе данных) — с операцией Commit()



Ваша диаграмма должна выглядеть, как показано на рис. 5.

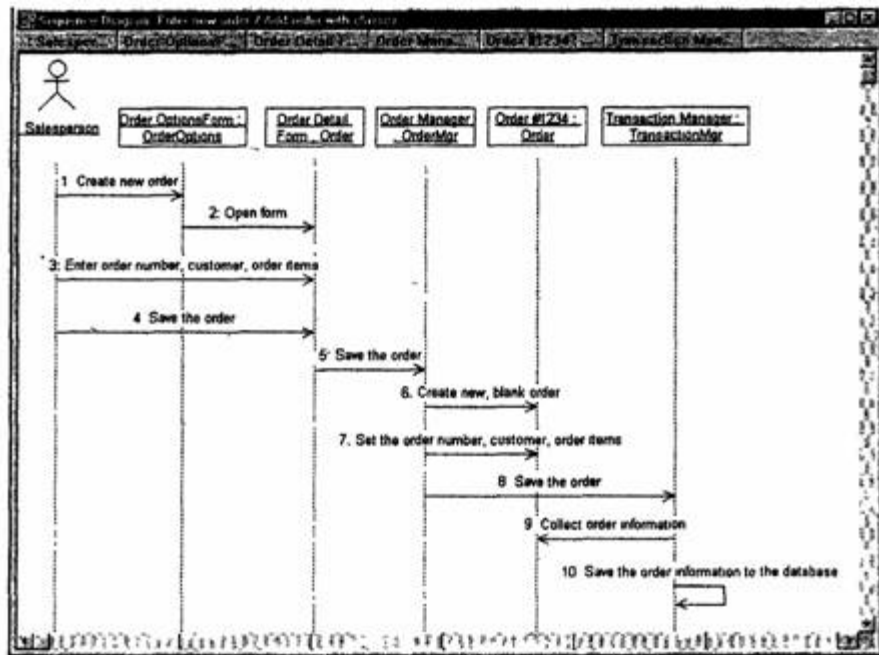


Рис. 4. Диаграмма Последовательности с именами классов

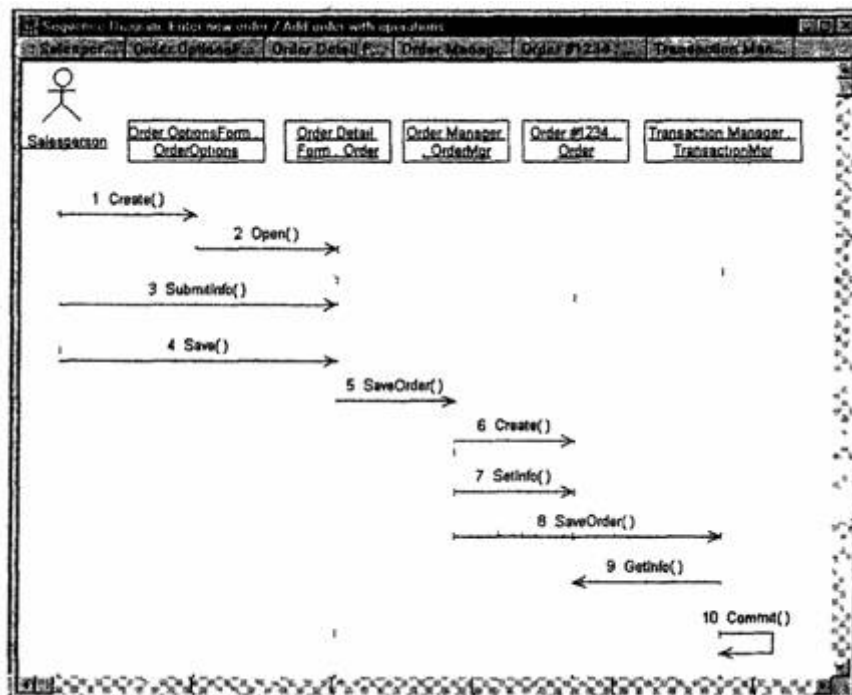


Рис. 5 — Диаграмма Последовательности с показанными на ней операциями

## **Создание Кооперативной диаграммы**

Для создания Кооперативной диаграммы достаточно нажать клавишу F5. Если же вы хотите сами проделать все требуемые операции, воспользуйтесь приводимым далее планом.

### **Создание Кооперативной диаграммы**

1. Щелкните правой кнопкой мыши на Логическом представлении в браузере
2. В открывшемся меню выберите пункт New > Collaboration Diagram (Создать > Кооперативная диаграмма).
3. Назовите эту диаграмму Add order (Ввод заказа).
4. Дважды щелкнув мышью на диаграмме, откройте ее.

### **Добавление действующего лица и объектов на диаграмму**

1. Перетащите действующее лицо Salesperson (Продавец) из браузера на диаграмму.
2. Нажмите кнопку Object (Объект) панели инструментов.
3. Щелкните мышью где-нибудь внутри диаграммы, чтобы поместить туда новый объект.
4. Назовите объект Order Options Form (Выбор варианта заказа).
5. Повторив шаги 3 и 4, поместите на диаграмму объекты:
  - Order Detail Form (Форма деталей заказа)
  - Order N1234 (Заказ Ns1234)

### **Добавление сообщений на диаграмму**

1. На панели инструментов нажмите кнопку Object Link (Связь объекта).
2. Проведите мышью от действующего лица Salesperson (Продавец) к объекту Order Options Form (Выбор варианта заказа).
3. Повторите шаги 1 и 2, соединив связями следующие объекты:
  - Действующее лицо Salesperson и объект Order Detail Form
  - Объект Order Options Form и объект Order Detail Form

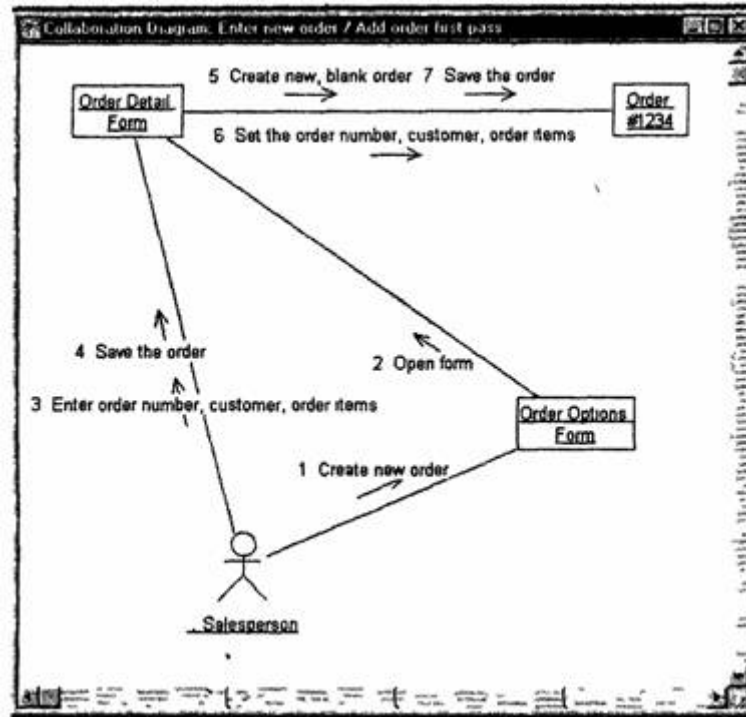
- Объект Order Detail Form и объект Order N1234
- 4. На панели инструментов нажмите кнопку Link Message (Собщение связи).
- 5. Щелкните мышью на связи между Salesperson и Order Options Form.
- 6. Выделив сообщение, введите его имя — Create new order (Создать новый заказ).
- 7. Повторив шаги с 4 по 6, поместите на диаграмму сообщения:
  - Open form (Открыть форму) — между Order Options Form и Order Detail Form
  - Enter order number, customer, order items (Ввести номер заказа, заказчика и число заказываемых предметов) — между Salesperson и Order Detail Form
  - Save the order (Сохранить заказ) — между Salesperson и Order Detail Form
  - Create new, blank order (Создать пустой заказ) — между Order Detail Form и Order N1234
  - Set the order number, customer, order items (Ввести номер заказа, заказчика и число заказываемых предметов) — между Order Detail Form и Order N1234
  - Save the order (Сохранить заказ) — между Order Detail Form и Order N1234

После завершения первого этапа работы Кооперативная диаграмма, описывающая ввод нового заказа, будет иметь вид, показанный на рис. 6. Теперь нужно поместить на диаграмму дополнительные элементы, а также рассмотреть ответственности объектов.

### **Добавление на диаграмму дополнительных объектов.**

1. Нажмите кнопку Object панели инструментов.
2. Щелкните мышью где-нибудь на диаграмме, чтобы поместить туда новый объект.
3. Введите имя объекта — Order Manager (Управляющий заказами).

4. На панели инструментов нажмите кнопку Object.
5. Поместите на диаграмму еще один объект.
6. Введите его имя — Transaction Manager (Управляющий транзакциями).



**Рис. 6 — Результат первого этапа работы над Кооперативной диаграммой, описывающей ввод нового заказа**

### Назначение ответственностей объектам

1. Выделите сообщение 5: Create new, blank order (Создать пустой заказ). Выделяйте слова, а не стрелку.
2. Нажав комбинацию клавиш CTRL+D, удалите это сообщение.
3. Повторите шаги 1 и 2 для удаления сообщений 6 и 7:
  - Set the order number, customer, order items
  - Save the order
4. Выделите связь между объектами Order Detail Form и Order N1234.
5. Нажав комбинацию клавиш CTRL+D, удалите эту связь.
6. На панели инструментов нажмите кнопку Object Link (Связь объекта).

7. Нарисуйте связь между Order Detail Form и Order Manager.
8. На панели инструментов нажмите кнопку Object Link (Связь объекта).
9. Нарисуйте связь между Order Manager и Order N1234.
10. На панели инструментов нажмите кнопку Object Link (Связь объекта).
11. Нарисуйте связь между Order N1234 и Transaction Manager.
12. На панели инструментов нажмите кнопку Object Link (Связь объекта).
13. Нарисуйте связь между Order Manager и Transaction Manager.
14. На панели инструментов нажмите кнопку Link Message (Сообщение связи).
15. Щелкните мышью на связи между объектами Order Detail Form и Order Manager, чтобы ввести новое сообщение.
16. Назовите это сообщение Save the order (Сохранить заказ).
17. Повторите шаги 14 — 16, добавив сообщения с шестого по девятое и назвав их:
  - Create new, blank order (Создать новый заказ) — между Order Manager и Order N1234
  - Set the order number, customer, order items (Ввести номер заказа, заказчика и число заказываемых предметов) — между Order Manager и Order N1234
  - Save the order (Сохранить заказ) — между Order Manager и Transaction Manager
  - Collect order information (Информация о заказе) — между Transaction Manager и Order N1'234
  - На панели инструментов нажмите кнопку Link to Self (Связь с собой).
  - Щелкнув на объекте Transaction Manager, добавьте к нему рефлексивное сообщение.
  - На панели инструментов нажмите кнопку Link Message (Сообщение связи).

- Щелкните мышью на рефлексивной связи Transaction Manager, чтобы ввести туда сообщение.
- Назовите новое сообщение Save the order information to the database (Сохранить информацию о заказе в базе данных).

Кооперативная диаграмма должна иметь вид, представленный на рис. 7.

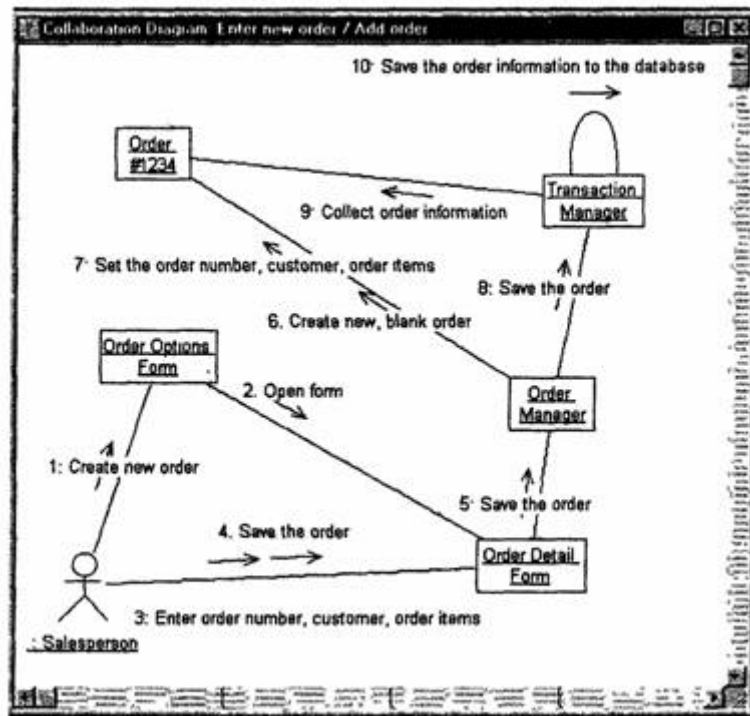


Рис. 7. Кооперативная диаграмма с дополнительными объектами

Соотнесение объектов с классами (если классы были созданы при разработке описанной выше диаграммы Последовательности)

1. Найдите в браузере класс Order Options.
2. Перетащите его на объект Order Options Form (Выбор варианта заказа) на диаграмме.
3. Повторите шаги 1 и 2, соотнесите остальные объекты и соответствующие им классы:

- Класс OrderDetail соотнесите с объектом Order Detail Form
- Класс OrderMgr — с объектом Order Manager
- Класс Order — с объектом Order N1234
- Класс TransactionMgr — с объектом Transaction Manager

Соотнесение объектов с классами (если вы не создавали опи-

санную выше диаграмму Последовательности)

1. Щелкните правой кнопкой мыши на объекте Order Options Form.
2. В открывшемся меню выберите пункт Open Specification (Открыть спецификацию).
3. В раскрывающемся списке классов выберите пункт (Создать). Появится окно спецификации классов.
4. В поле имени введите OrderOptions (Выбор заказа).
5. Щелкните на кнопке ОК. Вы вернетесь в окно спецификации объекта.
6. В списке классов выберите класс OrderOptions.
7. Щелкните на кнопке ОК, чтобы вернуться к диаграмме. Теперь объект называется Order Options Form : OrderOptions.
8. Для соотнесения остальных объектов с классами повторите шаги с 1 по 7:
  - Класс OrderDetail соотнесите с объектом Order Detail Form
  - Класс OrderMgr — с объектом Order Manager
  - Класс Order — с объектом Order N1234
  - Класс TransactionMgr — с объектом Transaction Manager .

Полученная к этому моменту Кооперативная диаграмма должна выглядеть, как показано на рис. 8.

Соотнесение сообщений с операциями (если операции были созданы при разработке описанной выше диаграммы Последовательности)

1. Щелкните правой кнопкой мыши на сообщении 1: Create new order (Создать новый заказ).
2. В открывшемся меню выберите пункт Open Specification (Открыть спецификацию).
3. В раскрывающемся списке имен укажите имя операции — Create() (Создать).
4. Нажмите на кнопку ОК.

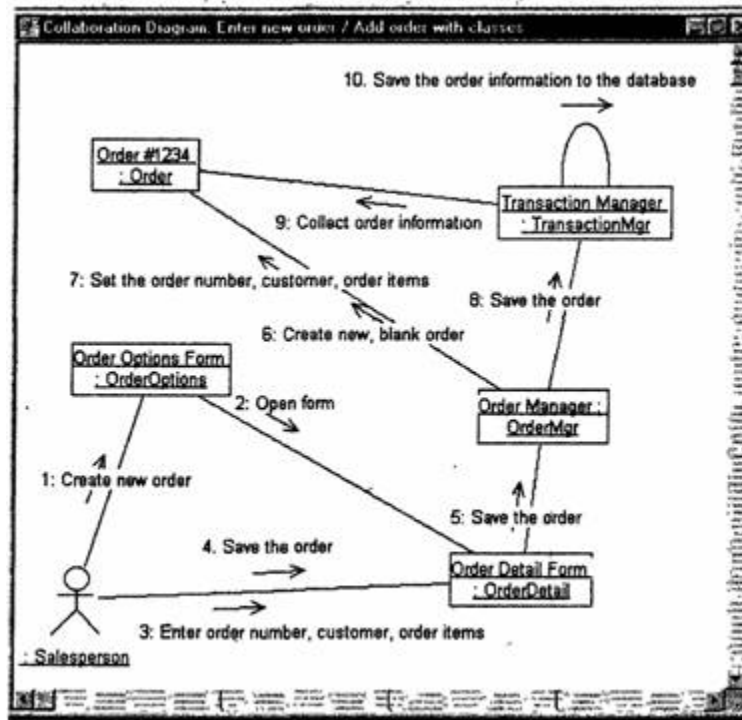


Рис. 8. Кооперативная диаграмма с именами классов

5. Повторите шаги 1 — 4 для соотнесения с операциями остальных сообщений:

- Сообщение 2: Open form (Открыть форму) соотнесите с операцией Open()
- Сообщение 3: Enter order number, customer, order items (Ввести номер заказа, заказчика и число заказываемых предметов) — с операцией SubmitInfoO
- Сообщение 4: Save the order (Сохранить заказ) — с операцией Save()
- Сообщение 5: Save the order (Сохранить заказ) — с операцией SaveOrder()
- Сообщение 6: Create new, blank order (Создать пустой заказ) — с операцией Create()
- Сообщение 7: Set the order number, customer, order items (Ввести номер заказа, заказчика и число заказываемых предметов) — с операцией SetInfo()
- Сообщение 8: Save the order (Сохранить заказ) — с операцией SaveOrder()



- Сообщение 9: Collect order information (Информация о заказе) — с операцией GetInfo()
- Сообщение 10: Save the order information to the database (Сохранить информацию о заказе в базе данных) — с операцией Commit()

Соотнесение сообщений с операциями (если вы не создавали описанную выше диаграмму Последовательности)

1. Щелкните правой кнопкой мыши на сообщении 1: Create new order (Создать новый заказ).
2. В открывшемся меню выберите пункт (создать операцию). Появится окно спецификации операции.
3. В поле имени введите имя операции — Create() (Создать).
4. Нажмите на кнопку ОК, чтобы закрыть окно спецификации операции и вернуться к диаграмме.
5. Еще раз щелкните правой кнопкой мыши на сообщении 1.
6. В открывшемся меню выберите пункт Open Specification (Открыть спецификацию).
7. В раскрывающемся списке Name <Имя> укажите имя новой операции.
8. Нажмите на кнопку ОК.
9. Повторите шаги 1 — 8, чтобы создать новые операции и соотнести с ними остальные сообщения:
  - Сообщение 2: Open form (Открыть форму) соотнесите с операцией Open()
  - Сообщение 3: Enter order number, customer, order items (Ввести номер заказа, заказчика и число заказываемых предметов) — с операцией SubmitInfo()
  - Сообщение 4: Save the order (Сохранить заказ) — с операцией Save()
  - Сообщение 5: Save the order (Сохранить заказ) — с операцией SaveOrderQ
  - Сообщение 6: Create new, blank order (Создать пустой заказ) — с операцией CreateO

- Сообщение 7: Set the order number, customer, order items (Ввести номер заказа, заказчика и число заказываемых предметов) — с операцией SettnfoQ
- Сообщение 8: Save the order (Сохранить заказ) — с операцией SaveOrderQ
- Сообщение 9: Collect order information (Информация о заказе) — с операцией GetinfoO
- Сообщение 10: Save the order information to the database (Сохранить информацию о заказе в базе данных) — с операцией Commit()

Ваша диаграмма должна выглядеть, как показано на рис. 9.

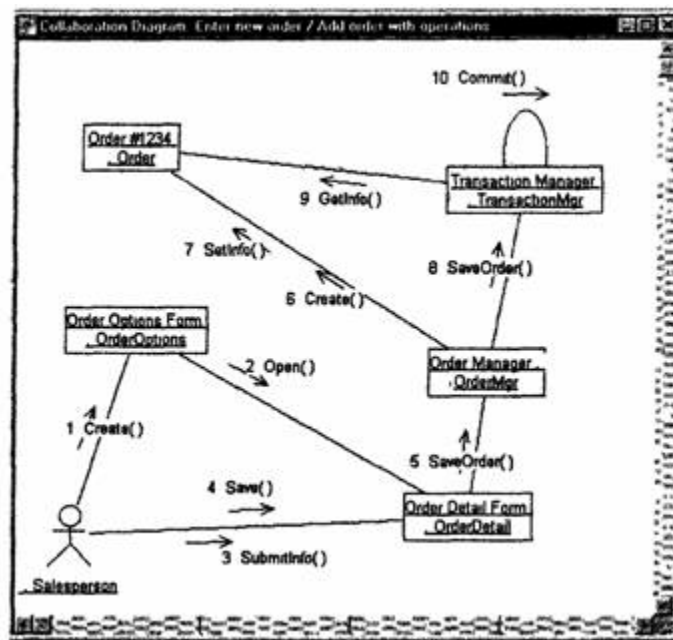


Рис. 9. Кооперативная диаграмма с показанными на ней операциями

## Практическое занятие № 3

### Классы и пакеты

Создайте диаграммы Классов для своего индивидуального задания.

#### Индивидуальное задание:

1. Создать диаграмму Классов
2. Сохранить файл модели, составить отчет

Пример выполнения упражнения описан ниже.

#### Упражнение

В этом упражнении необходимо сгруппировать в пакеты классы, созданные при выполнении предыдущего упражнения. Затем нужно будет построить несколько диаграмм Классов и показать на них классы и пакеты системы.

#### Создание диаграммы классов

Объедините обнаруженные нами классы в пакеты. Создайте диаграмму Классов для отображения пакетов, диаграмму Классов для представления классов в каждом пакете и диаграмму Классов для представления всех классов варианта использования “Ввести новый заказ”.

#### Этапы выполнения упражнения

##### Настройка

1. В меню модели выберите пункт Tools > Options (Инструменты > Параметры).
2. Перейдите на вкладку Diagram (Диаграмма).
3. Убедитесь, что установлен флажок Show stereotypes (Показать стереотипы).
4. Убедитесь, что установлены флажки Show All Attributes (Показать все атрибуты) и Show All Operations (Показать все операции).

5. Убедитесь, что сброшены флажки Suppress Attributes (Подавить вывод атрибутов) и Suppress Operations (Подавить вывод операций).

### Создание пакетов

1. Щелкните правой кнопкой мыши на Логическом представлении браузера.
2. В открывшемся меню выберите пункт New > Package (Создать > Пакет).
3. Назовите новый пакет Entities (Сущности).
4. Повторив шаги 1—3, создайте пакеты Boundaries (Границы) и Control (Управление). Браузер должен теперь иметь вид, показанный на рис. 1.

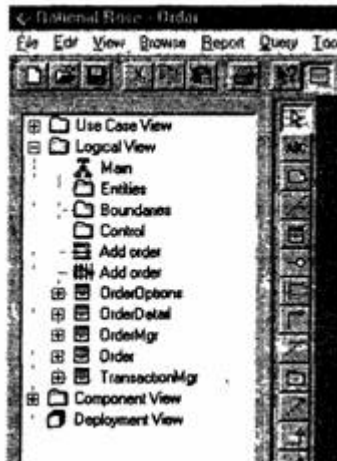


Рис. 1. Пакеты системы обработки заказов

### Создание Главной диаграммы Классов

1. Дважды щелкнув мышью на Главной диаграмме Классов, находящейся под Логическим представлением браузера, откройте ее.
2. Перетащите пакет Entities из браузера на диаграмму.
3. Перетащите пакеты Boundaries и Control из браузера на диаграмму.

Главная диаграмма Классов должна выглядеть, как показано на рис. 5.13.

Создание диаграммы Классов для сценария «Ввести новый за-

каз» с отображением всех классов

1. Щелкните правой кнопкой мыши на Логическом представлении браузера.
2. В открывшемся меню выберите пункт New > Class Diagram (Создать > Диаграмма Классов).
3. Назовите новую диаграмму Классов Add New Order (Ввод нового заказа).
4. Дважды щелкнув мышью на этой диаграмме в браузере; откройте ее.
5. Перетащите из браузера все классы (OrderOptions, OrderDetail, Order, OrderMgr и TransactionMgr). Полученная диаграмма Классов представлена на рис. 2.

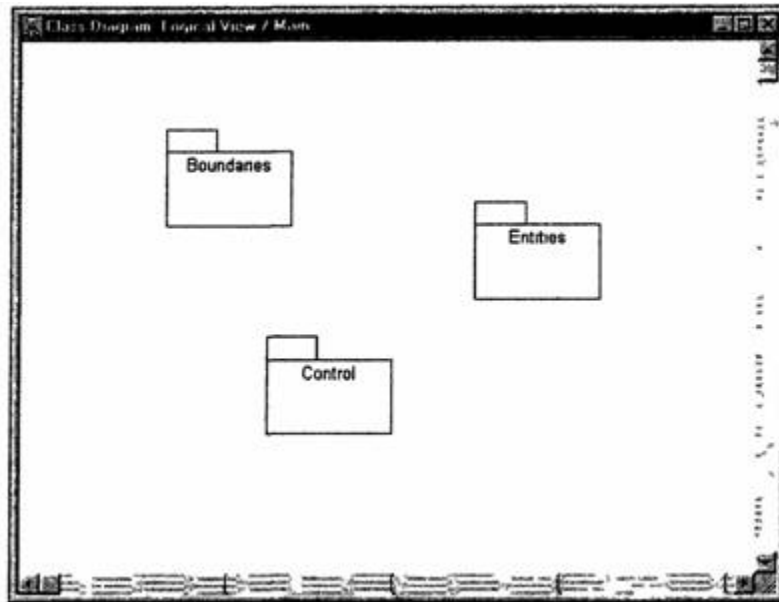


Рис. 2. Главная диаграмма Классов системы обработки заказов

### **Добавление стереотипов к классам**

1. Щелкните правой кнопкой мыши на классе OrderOptions диаграммы.
2. В открывшемся меню выберите пункт Open Specification (Открыть спецификацию).
3. В поле стереотипа введите слово Boundary.
4. Нажмите на кнопку ОК.

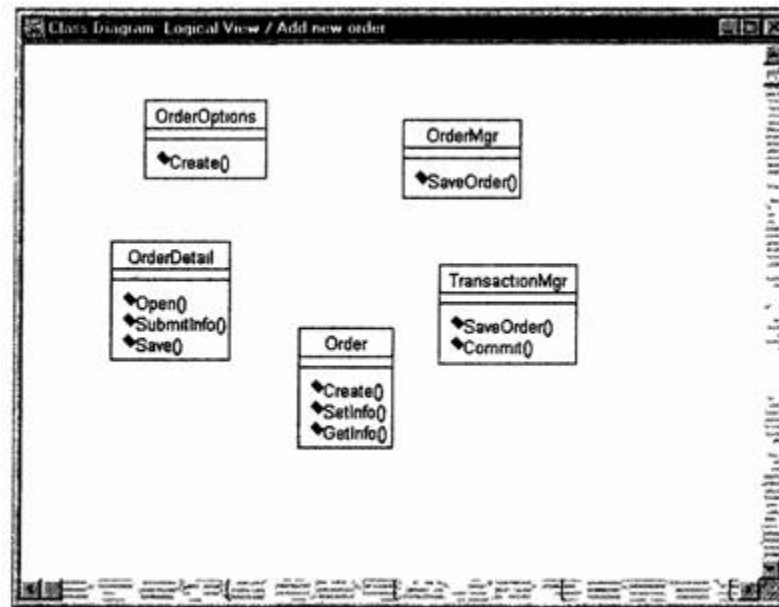


Рис. 3. Диаграмма Классов Add New Order

5. Щелкните правой кнопкой мыши на классе OrderDetail диаграммы.

6. В открывшемся меню выберите пункт Open Specification (Открыть спецификацию).

7. В раскрывающемся списке поля стереотипов будет указан стереотип Boundary. Выделите его

8. Нажмите на кнопку ОК.

9. Повторив шаги 1—4, свяжите классы OrderMgr и TransactionMgr со стереотипом Control, а класс Order — со стереотипом Entity

Теперь диаграмма Классов должна иметь вид, показанный на рис. 5.15.

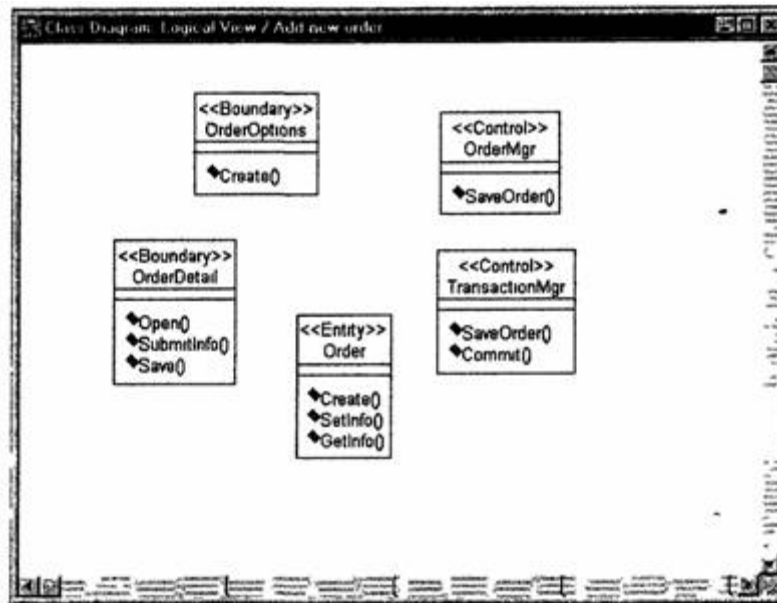


Рис. 4. Стереотипы классов для варианта использования «Ввести новый заказ»

### Объединение классов в пакеты

1. В браузере перетащите класс OrderOptions на пакет Boundaries.
2. Перетащите класс OrderDetail на пакет Boundaries.
3. Перетащите классы OrderMgr и TransactionMgr на пакет Control.
4. Перетащите класс Order на пакет Entities.

Классы и пакеты в браузере показаны на рис. 5.

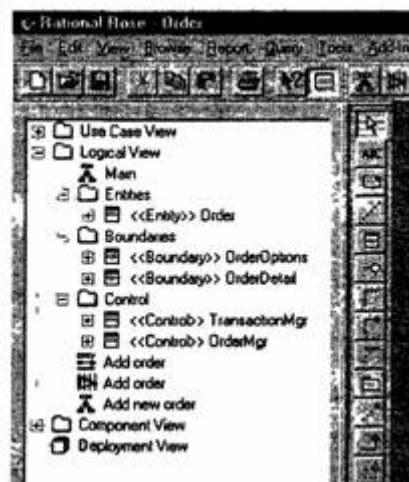


Рис. 5. Классы и пакеты варианта использования «Ввести новый заказ»

### Добавление диаграмм Классов к каждому пакету

1. В браузере Щелкните правой кнопкой мыши на пакете Boundaries.
2. В открывшемся меню выберите пункт New > Class Diagram (Создать > Диаграмма Классов).
3. Введите имя новой диаграммы — Main (Главная).
4. Дважды щелкнув мышью на этой диаграмме, откройте ее.
5. Перетащите на нее из браузера классы OrderOptions и OrderDetail. Главная диаграмма Классов пакета Boundaries должна теперь выглядеть, как показано на рис. 6.

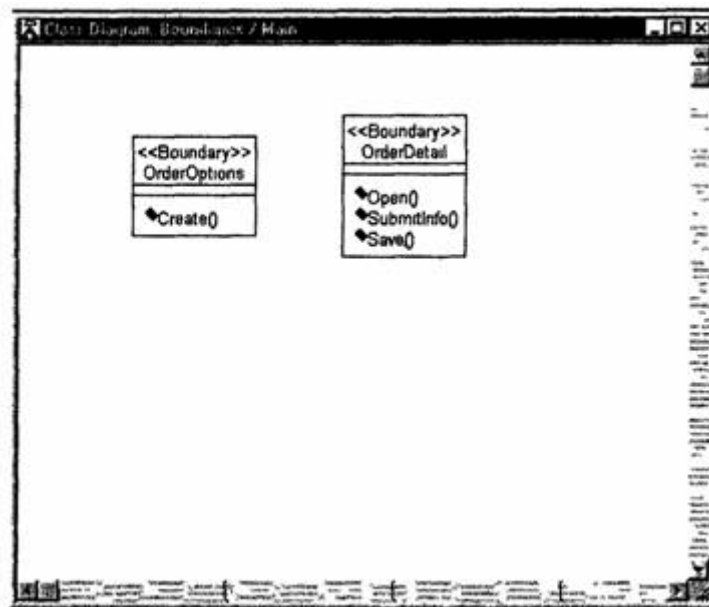


Рис. 6. Главная диаграмма Классов пакета Boundaries

6. Закройте диаграмму.
7. В браузере щелкните правой кнопкой мыши на пакете Entities.
8. В открывшемся меню выберите пункт New >- Class Diagram (Создать >- Диаграмма Классов).
9. Введите имя новой диаграммы — Main (Главная).
10. Дважды щелкнув мышью на этой диаграмме, откройте ее.



11. Перетащите на нее из браузера класс Order. Главная диаграмма Классов пакета Entities должна теперь иметь вид, представленный на рис. 7.



Рис. 7. Главная диаграмма Классов пакета Entities

12. Закройте диаграмму.

13. В браузере щелкните правой кнопкой мыши на пакете Control.

14. В открывшемся меню выберите пункт New > Class Diagram (Создать > Диаграмма Классов).

15. Введите имя новой диаграммы — Main (Главная).

16. Дважды щелкнув мышью на этой диаграмме, откройте ее

17. Перетащите на нее из браузера классы OrderMgr и TransactionMgr

18. Закройте диаграмму.

Главная диаграмма Классов пакета Control должна теперь выглядеть, как показано на рис 8.

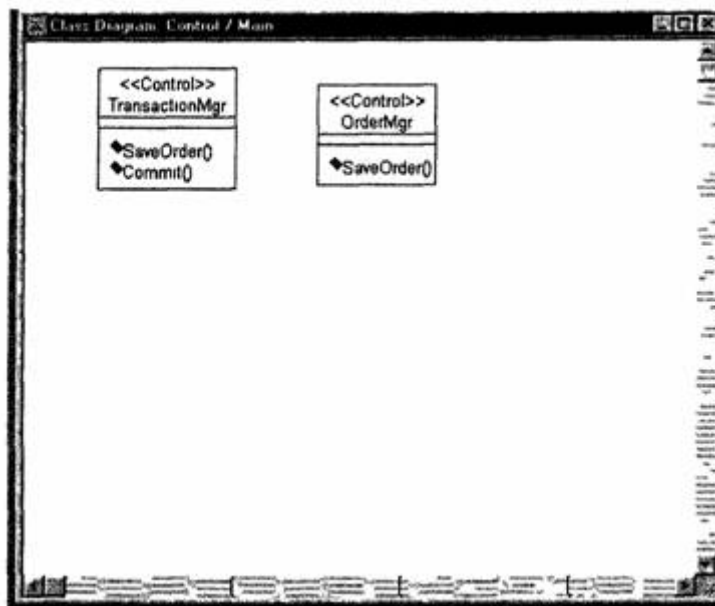


Рис. 8. Главная диаграмма Классов пакета Control

## Практическое занятие № 4

### Атрибуты и операции

В созданную диаграмму Классов для своего индивидуального задания добавьте атрибуты и операции.

#### Индивидуальное задание:

1. Модифицировать диаграмму Классов
  2. Сохранить файл модели, составить отчет
- Пример выполнения упражнения описан ниже.

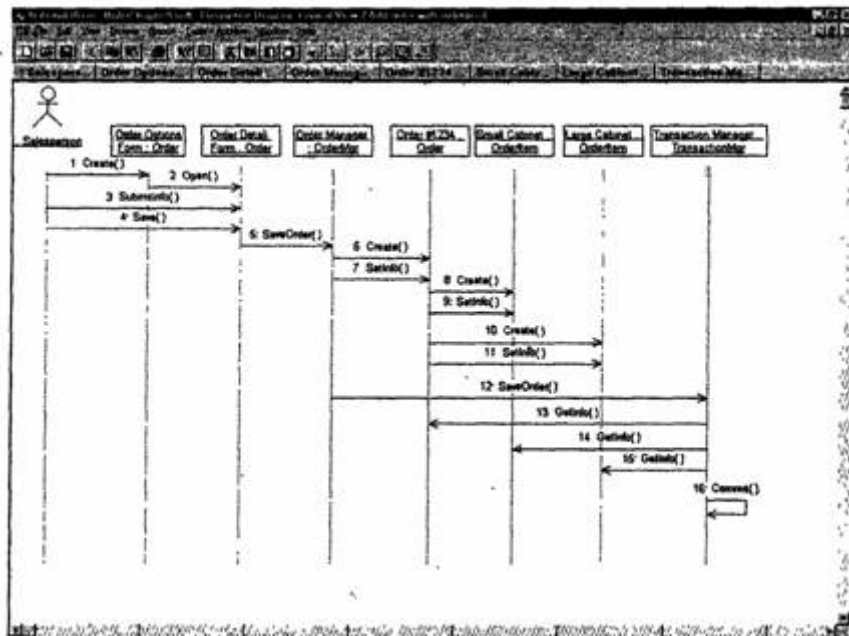


Рис. 1 — Модификация диаграммы Последовательности

#### Предыстория

Чтобы привести модель в соответствие с новыми идеями, пришлось обновить диаграмму Последовательности (см. рис. 1).

В этот момент Боб решил изменить требования: «Нам надо отслеживать дату заказа и дату его выполнения. Кроме того, так как у нас появились новые поставщики, слегка изменилась процедура инвентаризации».

Сначала Карен документировала новые требования относительно дат и рассмотрела изменения в процедуре инвентаризации

«на высоком уровне». Поскольку в данный момент она работала над вариантом использования «Ввести новый заказ», ее больше всего интересовало, как процедурные изменения повлияют на этот вариант использования.

Работа с вариантом использования «Провести инвентаризацию» была запланирована на следующий месяц, тогда она и позаботится о деталях соответствующих процедур. Оказалось, что изменения чрезвычайно сильно повлияют на вариант использования «Провести инвентаризацию», но совсем не отразятся на варианте использования «Ввести новый заказ».

Новые требования, связанные с датами, привели к тому, что пришлось добавить два атрибута в класс Order. После этого модель опять стала соответствовать последним предъявленным к системе требованиям.

### **Добавление атрибутов и операций**

Добавим атрибуты и операции к классам диаграммы Классов «Ввод нового заказа». При этом используем специфические для языка особенности. Установим параметры так, чтобы показывать все атрибуты, все операции и их сигнатуры. Применим нотацию UML.

### **Этапы выполнения упражнения**

#### **Настройка**

1. В меню модели выберите пункт Tools -> Options (Инструменты -> Параметры).
2. Перейдите на вкладку Diagram.
3. Убедитесь, что флажок Show visibility (Показать видимость) установлен.
4. Убедитесь, что флажок Show stereotypes (Показать стереотипы) установлен.
5. Убедитесь, что флажок Show operation signatures (Показать сигнатуры операций) установлен.

6. Убедитесь, что флажки Show all attributes (Показать все атрибуты) и Show all operations (Показать все операции) установлены.
7. Убедитесь, что флажки Suppress attributes (Подавить атрибуты) и Suppress operations (Подавить операции) сброшены.
8. Перейдите на вкладку Notation (Нотация).
9. Убедитесь, что флажок Visibility as icons (Отображать пиктограммы) сброшен.

### **Добавление нового класса**

1. Найдите в браузере диаграмму Классов варианта использования «Ввести новый заказ».
2. Дважды щелкнув мышью на диаграмме, откройте ее.
3. Нажмите кнопку Class панели инструментов.
4. Щелкните мышью внутри диаграммы? чтобы поместить туда новый класс.
5. Назовите его OrderItem.
6. Назначьте этому классу стереотип Entity.
7. В браузере перетащите класс в пакет Entities.

### **Добавление атрибутов**

1. Щелкните правой кнопкой мыши на классе Order.
2. В открывшемся меню выберите пункт New Attribute (Создать атрибут).
3. Введите новый атрибут: OrderNumber : Integer
4. Нажмите клавишу Enter.
5. Введите следующий атрибут: CustomerName : String.
6. Повторив шаги 4 и 5, добавьте атрибуты: OrderDate : Date  
OrderFUIDate : Date
7. Щелкните правой кнопкой мыши на классе OrderItem.
8. В открывшемся меню выберите пункт New Attribute (Создать атрибут).
9. Введите новый атрибут: ItemID : Integer.
10. Нажмите клавишу Enter.
11. Введите следующий атрибут: ItemDescription : String.

### **Добавление операций к классу OrderItem**

1. Щелкните правой кнопкой мыши на классе OrderItem.
2. В открывшемся меню выберите пункт New Operation (Создать операцию).
3. Введите новую операцию:Create.
4. Нажмите клавишу Enter.
5. Введите следующую операцию:SetInfo
6. Нажмите клавишу Enter.
7. Введите операцию:GetInfo

### **Подробное описание операций с помощью диаграммы**

#### **Классов**

1. Щелкнув мышью на классе Order, выделите его.
2. Щелкните на этом классе еще раз, чтобы переместить курсор внутрь.
3. Отредактируйте операцию Create(), чтобы она выглядела следующим образом:Create() : Boolean
4. Отредактируйте операцию SetInfo():SetInfo(OrderNum : Integer, Customer : String, OrderDate : Date, FillDate : Date) : Boolean
5. Отредактируйте операцию GetInfo(): GetInfo() : String

### **Подробное описание операций с помощью браузера**

1. Найдите в браузере класс OrderItem.
2. Раскройте этот класс, щелкнув на значке «+» рядом с ним В браузере появятся атрибуты и операции класса
3. Дважды щелкнув мышью на операции Oe11p1o(),откройте окно ее спецификации.
4. В раскрывающемся списке Return class (Возвращаемый класс) укажите String.
5. Щелкнув мышью на кнопке ОК, закройте окно спецификации операции
6. Дважды щелкните в браузере на операции SetInfo() класса OrderItem, чтобы открыть окно ее спецификации

7. В раскрывающемся списке Return class укажите Boolean.
8. Перейдите на вкладку Detail (Подробно).
9. Щелкните правой кнопкой мыши в области аргументов, чтобы добавить туда новый параметр
10. В открывшемся меню выберите пункт Insert (Вставить). Rose добавит аргумент под названием argname.
11. Щелкнув один раз на этом слове, выделите его и измените имя аргумента на ID.
12. Щелкните на колонке Type (Тип). В раскрывающемся списке типов выберите Integer.
13. Щелкните на колонке Default (По умолчанию), чтобы добавить значение аргумента по умолчанию. Введите число 0.
14. Нажав на кнопку ОК, закройте окно спецификации операции.
15. Дважды щелкните на операции Create() класса OrderItem, чтобы открыть окно ее спецификации.
16. В раскрывающемся списке Return class укажите Boolean.
17. Нажав на кнопку ОК, закройте окно спецификации операции.

### **Подробное описание операций**

1. Используя браузер или диаграмму Классов, введите следующие сигнатуры операций класса OrderDetail: Open() : Boolean  
SubmitInfo() : Boolean Save() : Boolean
2. Используя браузер или диаграмму Классов, введите сигнатуру операций класса OrderOptions: Create() : Boolean
3. Используя браузер или диаграмму Классов, введите сигнатуру операций класса OrderMgr: SaveOrder(OrderID : Integer): Boolean
4. Используя браузер или диаграмму Классов, введите сигнатуры операций класса TransactionMgr: SaveOrder(OrderID : Integer) : Boolean  
Commit() : Integer

## Практическое занятие № 5

### Связи

В созданной диаграмме Классов для своего индивидуального задания определить связи между классами.

#### Индивидуальное задание:

1. Модифицировать диаграмму Классов
2. Сохранить файл модели, составить отчет

Пример выполнения упражнения описан ниже.

В этом упражнении определяются связи между классами, участвующими в варианте использования «Ввести новый заказ».

#### Постановка задачи

После добавления к классам атрибутов и операций Карен была готова к генерации кода. Но сначала она должна была изучить связи между классами.

Чтобы найти связи, Карен просмотрела диаграммы Последовательности. Все взаимодействующие там классы нуждались в определении соответствующих связей на диаграммах Классов. После обнаружения связей Карен добавила их в модель.

#### Добавление связей

Добавим связи к классам, принимающим участие в варианте использования «Ввести новый заказ».

#### Этапы выполнения упражнения

##### Настройка

1. Найдите в браузере диаграмму Классов «Ввод нового заказа».
2. Дважды щелкнув на диаграмме, откройте ее.
3. Проверьте, имеется ли в панели инструментов диаграммы кнопка Unidirectional Association (Однонаправленная ассоциация).



Если ее нет, продолжите настройку, выполнив шаги 4 и 5. Если есть, приступайте к выполнению самого упражнения.

4. Щелкните правой кнопкой мыши на панели инструментов диаграммы и в открывшемся меню выберите пункт **Customize** (Настроить).

5. Добавьте на панель кнопку **Creates A Unidirectional Association** (Создать однонаправленную ассоциацию).

### **Добавление ассоциаций**

1. Нажмите кнопку **Unidirectional Association** панели инструментов.

2. Проведите ассоциацию от класса **OrderOptions** к классу **OrderDetail**.

3. Повторите шаги 1 и 2, создав ассоциации:

- От класса **OrderDetail** к классу **OrderMgr**
- От класса **OrderMgr** к классу **Order**
- От класса **OrderMgr** к классу **TransactionMgr**
- От класса **TransactionMgr** к классу **Order**
- От класса **TransactionMgr** к классу **OrderItem**
- От класса **Order** к классу **OrderItem**

4. Щелкните правой кнопкой мыши на однонаправленной ассоциации между классами **OrderOptions** и **OrderDetail** со стороны класса **OrderOptions**.

5. В открывшемся меню выберите пункт **Multiplicity > Zero or One** (Множественность >- Нуль или один).

6. Щелкните правой кнопкой мыши на другом конце однонаправленной ассоциации.

7. В открывшемся меню выберите пункт **Multiplicity > Zero or One** (Множественность >- Нуль или один).

8. Повторите шаги 4 — 7, добавив на диаграмму значения множественности для остальных ассоциаций, как показано на рис. 1.

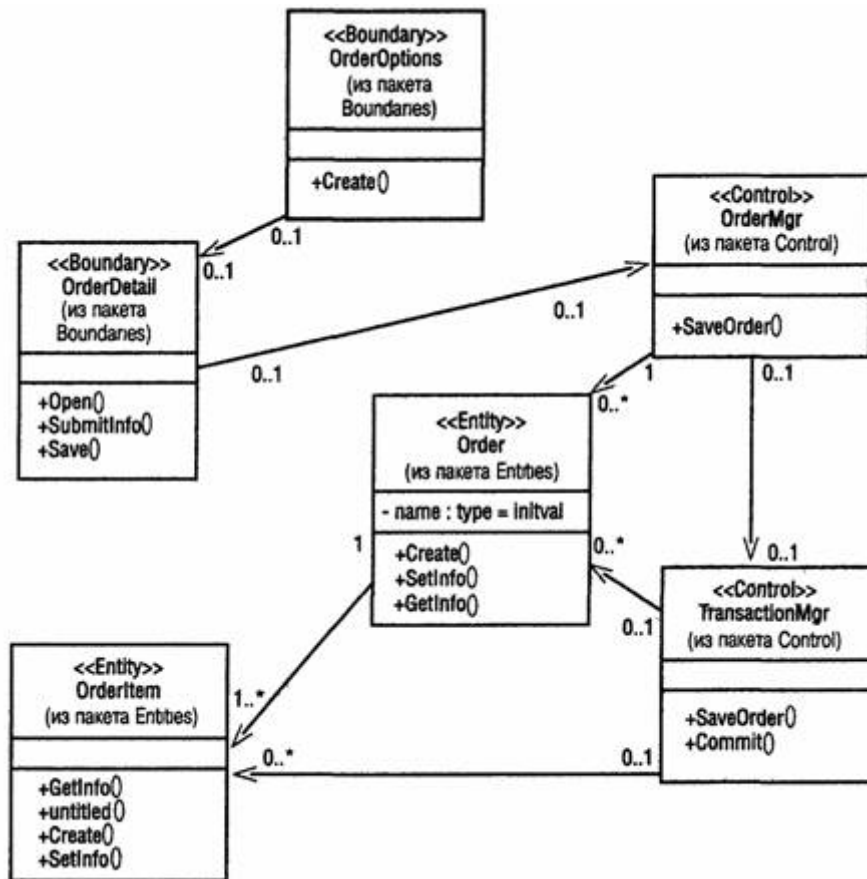


Рис. 1. Ассоциации сценария «Ввести новый заказ»

## Практическое занятие № 6

### Поведение объектов

Создать диаграмму состояний для своего индивидуального задания.

#### Индивидуальное задание:

1. Создать диаграмму Состояний
2. Сохранить файл модели, составить отчет

Пример выполнения упражнения описан ниже.

В этом упражнении создается диаграмма Состояний для класса Order.

#### Пример

##### Постановка задачи

Проектируя класс Order, Карен поняла, что за его поведением нужно наблюдать. Многие требования к классу значительно изменяются при изменении состояния его экземпляра. Например, заказы, выполнение которых приостановлено, ведут себя не так, как выполненные заказы, а те в свою очередь не так, как отмененные заказы.

Для того чтобы убедиться, что проект удовлетворяет всем этим требованиям, Карен со своей группой разработчиков создала диаграмму Состояний для класса Order. С ее помощью разработчики смогли понять, как писать код для этого класса.

##### Создание диаграммы Состояний

Постройте диаграмму Состояний для класса Order, показанную на рис. 1.

#### Этапы выполнения упражнения

##### Создание диаграммы

1. Найдите в браузере класс Order.

2. Щелкните на классе правой кнопкой мыши и в открывшемся меню укажите пункт Open State Diagram (Открыть диаграмму состояний).

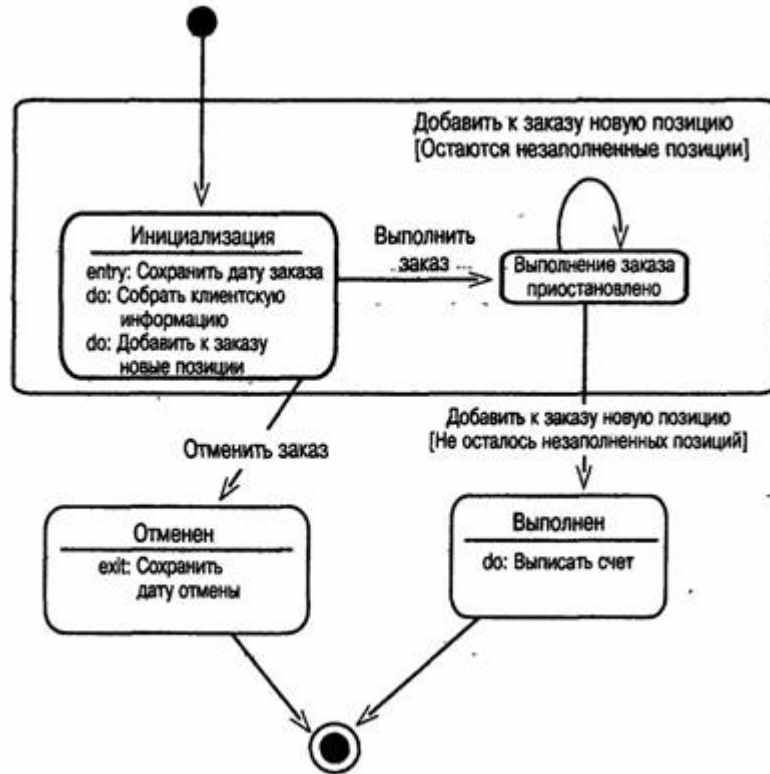


Рис. 1 — Диаграмма Состояний для класса Order

### Добавление начального и конечного состояний

1. Нажмите кнопку Start State (Начальное состояние) панели инструментов.
2. Поместите это состояние на диаграмму.
3. Нажмите кнопку End State (Конечное состояние) панели инструментов.
4. Поместите это состояние на диаграмму.

### Добавление суперсостояния

1. Нажмите кнопку State (Состояние) панели инструментов.
2. Поместите это состояние на диаграмму.

### Добавление оставшихся состояний

1. На панели инструментов нажмите кнопку State (Состояние).

2. Поместите состояние на диаграмму.
3. Назовите состояние Cancelled (Отменен).
4. На панели инструментов нажмите кнопку State (Состояние).
5. Поместите состояние на диаграмму.
6. Назовите состояние Filled (Выполнен).
7. На панели инструментов нажмите кнопку State (Состояние).
8. Поместите состояние на диаграмму внутрь суперсостояния.
9. Назовите состояние Initialization (Инициализация).
10. На панели инструментов нажмите кнопку State (Состояние).
11. Поместите состояние на диаграмму внутрь суперсостояния.
12. Назовите состояние Pending (выполнение заказа приостановлено).

### **Описание состояний**

1. Дважды щелкните мышью на состоянии Initialization (Инициализация).
2. Перейдите на вкладку Detail (Подробно).
3. Щелкните правой кнопкой мыши в окне Actions (Действия).
4. В открывшемся меню выберите пункт Insert (Вставить).
5. Дважды щелкните мышью на новом действии.
6. Назовите его Store order date (Сохранить дату заказа).
7. Убедитесь, что в окне When (Когда) указан пункт On Entry (На входе).
8. Повторив шаги 3—7, добавьте следующие действия:
  - Collect customer info (Собрать клиентскую информацию), в окне When укажите Entry until Exit (Выполнять до завершения)
  - Add order items (Добавить к заказу новые позиции), укажите Entry until Exit (Выполнять до завершения)
9. Нажмите два раза на ОК, чтобы закрыть спецификацию.
10. Дважды щелкните мышью на состоянии Cancelled (Отменен).
11. Повторив шаги 2 — 7, добавьте действие: • Store cancellation data (Сохранить дату отмены), укажите On Exit (На выходе)
12. Нажмите два раза на ОК, чтобы закрыть спецификацию.

13. Дважды щелкните мышью на состоянии Filled (Выполнен).
14. Повторив шаги 2—7, добавьте действие: • Bill customer (Выписать счет), укажите Entry until Exit
15. Нажмите два раза на ОК, чтобы закрыть спецификацию.

### **Добавление переходов**

1. Нажмите кнопку Transition (Переход) панели инструментов.
2. Щелкните мышью на начальном состоянии.
3. Проведите линию перехода к состоянию Initialization (Инициализация).
4. Повторив шаги с первого по третий, создайте следующие переходы:
  - От состояния Initialization (Инициализация) к состоянию Pending (Выполнение заказа приостановлено)
  - От состояния Pending (Выполнение заказа приостановлено) к состоянию Filled (Выполнен)
  - От суперсостояния к состоянию Cancelled (Отменен)
  - От состояния Cancelled (Отменен) к конечному состоянию
  - От состояния Filled (Выполнен) к конечному состоянию
5. На панели инструментов нажмите кнопку Transition to Self (Переход к себе).
6. Щелкните мышью на состоянии Pending (Выполнение заказа приостановлено).

### **Описание переходов**

1. Дважды щелкнув мышью на переходе от состояния Initialization (Инициализация) к состоянию Pending (Выполнение заказа приостановлено), откройте окно спецификации перехода.
2. В доле Event (Событие) введите фразу Finalize order (Выполнить заказ).
3. Щелкнув на кнопке ОК, закройте окно спецификации.
4. Повторив шаги с первого по третий, добавьте событие Cancel Order (Отменить заказ) к переходу между суперсостоянием и состоянием Cancelled (Отменен).

5. Дважды щелкнув мышью на переходе от состояния Pending (Выполнение заказа приостановлено) к состоянию Filled (Выполнен), откройте окно его спецификации.

6. В поле Event (Событие) введите фразу Add order item (Добавить к заказу новую позицию).

7. Перейдите на вкладку Detail (Подробно).

8. В поле Condition (Условие) введите No unfilled items remaining (Не осталось незаполненных позиций).

9. Щелкнув на кнопке ОК, закройте окно спецификации.

10. Дважды щелкните мышью на рефлексивном переходе (Transition to Self) состояния Pending (Выполнение заказа приостановлено).

11. В поле Event (Событие) введите фразу Add order item (Добавить к заказу новую позицию).

12. Перейдите на вкладку Detail (Подробно).

13. В поле Condition (Условие) введите Unfilled items remaining (Остаются незаполненные позиции).

14. Щелкнув на кнопке ОК, закройте окно спецификации.

## Практическое занятие № 7

### Представление компонентов

Создать диаграмму компонентов для своего индивидуального задания.

#### Индивидуальное задание:

1. Создать диаграмму Компонентов
2. Сохранить файл модели, составить отчет

Пример выполнения упражнения описан ниже.

В этом упражнении создается диаграмма Компонентов системы обработки заказов. На данный момент уже определены все классы, требуемые для варианта использования «Ввести новый заказ». По мере реализации других вариантов использования на диаграмму следует добавлять новые компоненты.

#### Пример

##### Постановка задачи

По завершении анализа и проектирования системы Дэн, один из разработчиков проекта, построил диаграммы Компонентов. Выбрав в качестве языка программирования C++, Дэн создал для каждого класса соответствующие этому языку компоненты.

На рис. 1 показана главная диаграмма Компонентов системы. Внимание на ней уделяется пакетам создаваемых компонентов.

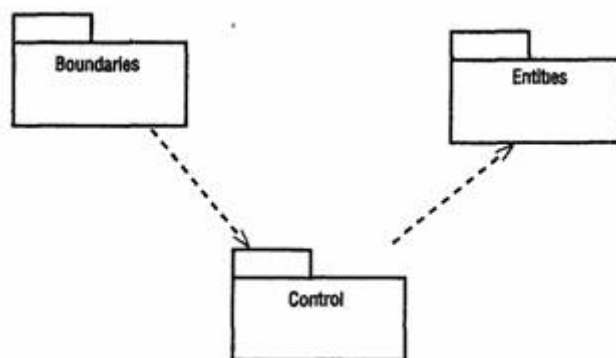


Рис. 1 — Главная диаграмма Компонентов системы



На рис. 2 изображены все компоненты пакета Entities. Они содержат классы пакета Entities Логического представления системы.

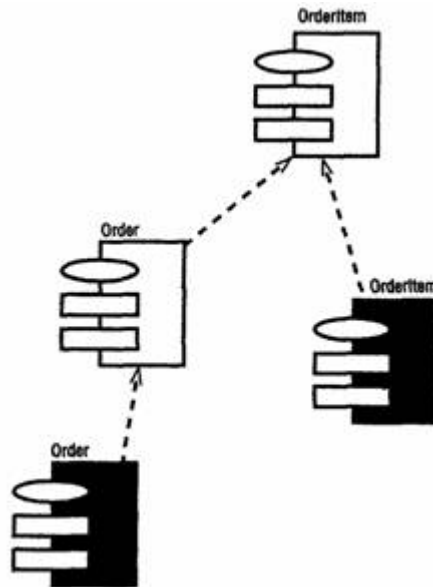


Рис. 2 — Диаграмма Компонентов пакета Entities

На рис. 3 показаны компоненты пакета Control. Они содержат классы пакета Control Логического представления системы.

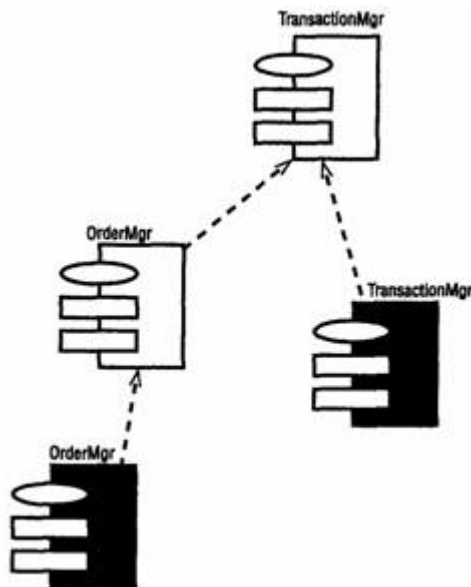


Рис. 3 — Диаграмма Компонентов пакета Control

Наконец, на рис. 4 показаны компоненты пакета Boundaries. Они также соответствуют классам одноименного пакета Логического представления системы.

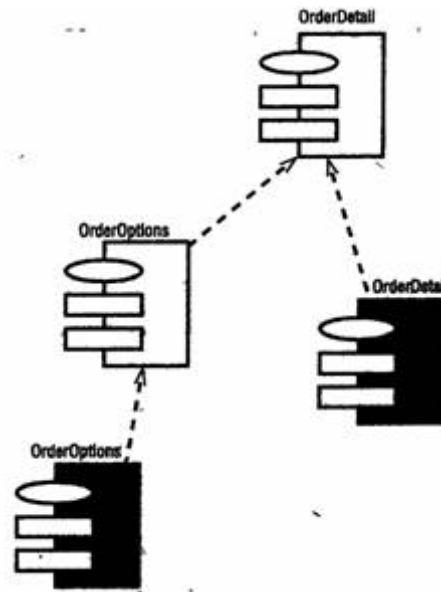


Рис. 4 — Диаграмма Компонентов пакета Boundaries

На рис. 5 показаны все компоненты системы, на этой диаграмме можно видеть все зависимости между всеми компонентами проектируемой системы.

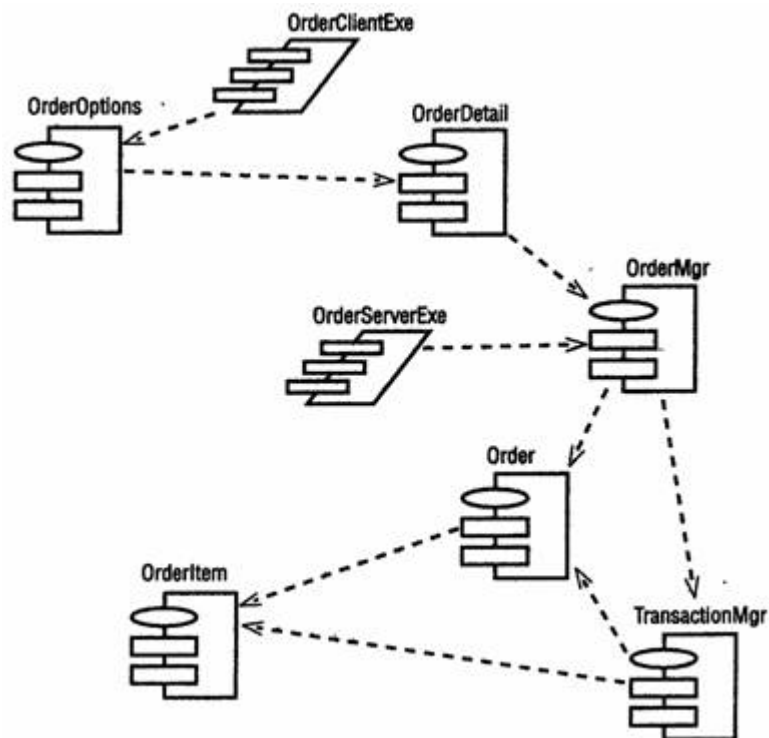


Рис. 5 — Диаграмма Компонентов системы

## Этапы выполнения упражнения

### **Создание пакетов компонентов**

1. Щелкните правой кнопкой мыши на представлении компонентов в браузере.
2. В открывшемся меню выберите пункт New > Package (Создать > Пакет).
3. Назовите пакет Entities (Сущности).
4. Повторив шаги с первого по третий, создайте пакеты Boundaries (Границы) и Control (Управление).

### **Добавление пакетов на Главную диаграмму Компонентов**

1. Откройте Главную диаграмму Компонентов, дважды щелкнув на ней мышью.
2. Перетащите пакеты Entities, Boundary и Control из браузера на Главную диаграмму.

### **Отображение зависимостей между пакетами**

1. Нажмите кнопку Dependency (Зависимость) панели инструментов.
2. Щелкните мышью на пакете Boundaries Главной диаграммы Компонентов.
3. Проведите линию зависимости к пакету Control.
4. Повторив шаги 1 — 3, проведите зависимость от пакета Control к пакету Entities.

### **Добавление компонентов к пакетам и отображение зависимостей**

1. Дважды щелкнув мышью на пакете Entities Главной диаграммы Компонентов, откройте Главную диаграмму Компонентов этого пакета.
2. Нажмите кнопку Package Specification (Спецификация пакета) панели инструментов.
3. Поместите спецификацию пакета на диаграмму.

4. Введите имя спецификации пакета — OrderItem.
  5. Повторив шаги 2—4, добавьте спецификацию пакета Order.
  6. Нажмите кнопку Package Body (Тело пакета) панели инструментов.
  7. Поместите его на Диаграмму.
  8. Введите имя тела пакета — OrderItem.
  9. Повторив шаги 6—8, добавьте тело пакета Order.
  10. Нажмите кнопку Dependency (Зависимость) панели инструментов.
  11. Щелкните мышью на теле пакета OrderItem.
  12. Проведите линию зависимости к спецификации пакета OrderItem.
  13. Повторив шаги 10 — 12, добавьте линию зависимости между телом пакета Order и спецификацией пакета Order.
  14. Повторив шаги 10 — 12, добавьте линию зависимости от спецификации пакета Order к спецификации пакета OrderItem.
  15. С помощью описанного метода создайте следующие компоненты и зависимости:
- Для пакета Boundaries:
- Спецификацию пакета OrderOptions
  - Тело пакета OrderOptions
  - Спецификацию пакета OrderDetail
  - Тело пакета OrderDetail Зависимости в пакете Boundaries:
  - От тела пакета OrderOptions к спецификации пакета OrderOptions
  - От тела пакета OrderDetail к спецификации пакета OrderDetail
  - От спецификации пакета OrderOptions к спецификации пакета OrderDetail
- Для пакета Control:
- Спецификацию пакета OrderMgr
  - Тело пакета OrderMgr
  - Спецификацию пакета TransactionMgr
  - Тело пакета TransactionMgr

Зависимости в пакете Control:

- От тела пакета OrderMgr к спецификации пакета OrderMy
- От тела пакета TransactionMgr к спецификации пакета TransactionMgr
- От спецификации пакета OrderMgr к спецификации пакета TransactionMgr

### **Создание диаграммы Компонентов системы**

1. Щелкните правой кнопкой мыши на представлении Компонентов в браузере.
2. В открывшемся меню выберите пункт New > Component Diagram (Создать > Диаграмма Компонентов).
3. Назовите новую диаграмму System.
4. Дважды щелкните на этой диаграмме мышью.

### **Размещение компонентов на диаграмме Компонентов системы**

1. Разверните в браузере пакет компонентов Entities, чтобы открыть его.
2. Щелкните мышью на спецификации пакета Order в пакете компонентов Entities.
3. Перетащите эту спецификацию на диаграмму.
4. Повторив шаги 2 и 3, поместите на диаграмму спецификацию пакета OrderItem.
5. С помощью этого метода поместите на диаграмму следующие компоненты:

Из пакета компонентов Boundaries:

- Спецификацию пакета OrderOptions
- Спецификацию пакета OrderDetails Из пакета компонентов

Control:

- Спецификацию пакета OrderMgr
- Спецификацию пакета TransactionMgr
- Спецификацию задачи OrderContentExec
- Спецификацию задачи OrderServerExec

6. Нажмите кнопку Task Specification (Спецификация задачи) панели инструментов.

7. Поместите на диаграмму спецификацию задачи и назовите ее OrderClientExe.

8. Повторите шаги 6 и 7 для спецификации задачи OrderServerExe.

### **Добавление оставшихся зависимостей на диаграмму Компонентов системы**

Уже существующие зависимости будут автоматически показаны на диаграмме Компонентов системы после добавления туда соответствующих компонентов. Теперь нужно добавить остальные зависимости.

1. Нажмите кнопку Dependency (Зависимость) панели инструментов.

2. Щелкните мышью на спецификации пакета OrderDetail.

3. Проведите линию зависимости к спецификации пакета OrderMgr.

4. Повторив шаги 1—3, создайте следующие зависимости:

- От спецификации пакета OrderMgr к спецификации пакета Order

- От спецификации пакета TransactionMgr к спецификации пакета OrderItem

- От спецификации пакета TransactionMgr к спецификации пакета Order

- От спецификации задачи OrderClientExe к спецификации пакета OrderOptions

- От спецификации задачи OrderServerExe к спецификации пакета OrderMgr

## **Практическое занятие № 8**

### **Представление размещения**

Создать диаграмму размещения для своего индивидуального задания.

#### **Индивидуальное задание:**

1. Создать диаграмму Размещения
2. Сохранить файл модели, составить отчет

Пример выполнения упражнения описан ниже.

В этом примере создается диаграмма Размещения системы обработки заказов.

#### **Пример**

##### **Постановка задачи**

Команда разработчиков завершила все предшествующие данному моменту этапы анализа и проектирования системы. Варианты использования, взаимодействие между объектами и компоненты четко описаны. Тем не менее, подразделению администрирования сети необходимо знать, на каких компьютерах будут размещаться различные компоненты системы. В связи с этим пришлось разработать диаграмму Размещения для системы обработки заказов.

#### **Создание диаграммы Размещения**

Разработайте диаграмму Размещения для системы обработки заказов. Готовая диаграмма показана на рис. 1.

##### **Этапы выполнения упражнения**

#### **Добавление узлов к диаграмме Размещения**

1. Дважды щелкнув мышью на представлении Размещения в браузере, откройте диаграмму Размещения.
2. Нажмите кнопку Processor (Процессор) панели инструментов.

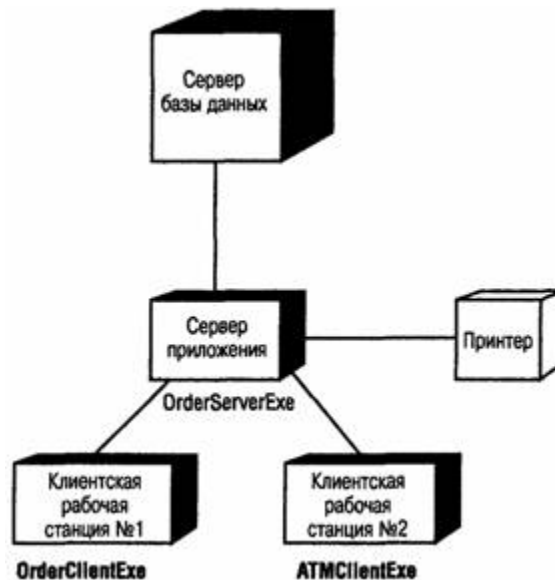


Рис. 1 — Диаграмма Размещения для системы обработки заказов

3. Щелкнув мышью на диаграмме, поместите туда процессор.
4. Введите имя процессора «Сервер базы данных».
5. Повторив шаги 2—4, добавьте следующие процессоры:
  - Сервер приложения
  - Клиентская рабочая станция №1
  - Клиентская рабочая станция №2
6. На панели инструментов нажмите кнопку Device (Устройство).
7. Щелкнув мышью на диаграмме, поместите туда устройство.
8. Назовите его «Принтер».

### Добавление связей

1. Нажмите кнопку Connection (Связь) панели инструментов.
2. Щелкните мышью на процессоре «Сервер базы данных».
3. Проведите линию связи к процессору «Сервер приложения».
4. Повторив шаги 1—3, добавьте следующие связи;
  - От процессора «Сервер приложения» к процессору «Клиентская рабочая станция №1»
  - От процессора «Сервер приложения» к процессору «Клиентская рабочая станция №2»
  - От процессора «Сервер приложения» к устройству «Принтер»



### **Добавление процессов**

1. Щелкните правой кнопкой мыши на процессоре «Сервер приложения» в браузере.
2. В открывшемся меню выберите пункт New > Process (Создать > Процесс).
3. Введите имя процесса — OrderServerExe.
4. Повторив шаги 1—3, добавьте процессы:
  - Процесс OrderClientExe на процессоре «Клиентская рабочая станция №1»
  - Процесс ATMClientExe на процессоре «Клиентская рабочая станция №2»

### **Показ процессов на диаграмме**

1. Щелкните правой кнопкой мыши на процессоре «Сервер приложения».
2. В открывшемся меню выберите пункт Show Processes (Показать процессы).
3. Повторив шаги 1 и 2, покажите процессы на следующих процессорах:
  4. Клиентская рабочая станция №1
  5. Клиентская рабочая станция №2

## СПИСОК ЛИТЕРАТУРЫ

1. Схиртладзе А. Г. Интегрированные системы проектирования и управления: учебник. — М.: Академия, 2010.
2. Кознов Д.В., Бугайченко Д.Ю. Введение в программную инженерию. — М.: ИНТУИТ.РУ «Интернет-университет информационных технологий», 2009.
3. Кудрявцев Е.М. Основы автоматизированного проектирования: учебник. — М. Академия, 2011.
4. Грекул В.И. Проектирование информационных систем. — М. : ИНТУИТ.РУ «Интернет-университет информационных технологий», 2008.
5. Кузнецов С.Д. Базы данных. — М.: Академия, 2012.
6. Теория информационных процессов и систем / Под ред. Советова Б.Я. — М.: Академия, 2011.
7. Черников Б. В. Информационные технологии управления. — М. : ИНФРА-М, 2011.
8. Моделирование систем / С.И. Дворецкий [и др.]. — М. : Академия, 2009.
9. Советов Б.Я. Моделирование систем. — М. : Юрайт, 2012.
10. Морозов В. К. Моделирование информационных и динамических систем. — М.: Академия, 2011.
11. Павловский Ю. Н. Имитационное моделирование. — М.: Академия, 2008.
12. Кондаков А. И. САПР технологических процессов: учебник. — М.: Академия, 2007.
13. Мельников В.П. Информационные технологии. — М.: Академия, 2012.
14. Анашкина Н. В. Технология и методы программирования. — М.: Академия, 2012.
15. Смоленцев В. П. Управление системами и процессами. — М.: Академия, 2010.

## ПРИЛОЖЕНИЕ

### Варианты заданий

#### Вариант 1. Комитет по имуществу

##### *Описание:*

В городском комитете по имуществу имеются сведения о нежилом фонде и факте приватизации некоторых зданий. Каждое здание имеет регистрационный номер, характеризуется адресом с указанием региона, площадью, количеством этажей, датой постройки. Приватизация проводится на аукционе нежилого фонда. Одно и то же здание может выставляться на нескольких аукционах. На одном аукционе – несколько зданий. Аукцион характеризуется датой проведения, организацией, проводящей аукцион, адресом проведения аукциона. Каждое приватизированное здание характеризуется датой продажи, первоначальной аукционной стоимостью, конечной стоимостью, данными о покупателе. Данные о покупателе включают для организации: ИНН, название, ФИО директора, адрес.

##### *Задания:*

1. Составить отчет о приватизированных зданиях 110 микрорайона г. Невинномыска, приватизированных на аукционе 20.05.2003 г., проводимой городской администрацией, с указанием регистрационного номера, даты постройки, первоначальной аукционной стоимостью, конечной стоимостью.
2. Какие здания были приватизированы в N месяце?
3. Сколько зданий приватизировано в N году и на какую общую сумму?
4. На каком аукционе приватизировано наибольшее количество зданий?

#### Вариант 2. Фирмы по продаже компьютеров

**Описание:**

В городской администрации хранится информация о фирмах, занимающихся распространением компьютерной техники. Каждая фирма характеризуется названием, телефоном, факсом, адресом, ФИО директора. Кроме этого каждая фирма характеризуется формой собственности (государственная, ТОО, частная, акционерная, городская), общей площадью помещения, общим числом сотрудников.

Каждая фирма имеет сотрудников, характеризующихся ФИО, должностью, окладом.

У каждой фирмы есть прайс-лист по имеющемуся в продаже товару. В прайс листе указывается: шифр товара, наименование, цена, срок гарантии.

**Задания:**

1. Составить перечень организаций с указанием телефона и ФИО директора частной формы собственности, у которых мониторы СТХ имеют гарантию более 1 года.
2. Составить перечень организаций с указанием телефона и ФИО директора государственной формы собственности, у которых оклад главного менеджера больше 500 р.
3. Определить фирму, у которой средний возраст сотрудников минимальный.

**Вариант 3. Профсоюзная организация****Описание:**

В профсоюзной организации хранится информация о членах профсоюза (номер профсоюзного билета, ФИО, стаж работы, дата вступления в профсоюз), а также о путевках и материальной помощи, которыми пользовались члены профсоюзной организации. О каждой путевке хранится следующая информация: Тип (лечебная, туристическая), наименование организации, срок действия (24 дня, 12 дней), дата с которой действует данная путевка. О материальной

помощи хранится следующая информация: дата выдачи, сумма, основание для оказания материальной помощи.

***Задания:***

1. Составить список членов профсоюза с указанием стажа работы и даты вступления в профсоюз, которые пользовались материальной помощью на лечение в 2002 году.

2. Составить перечень лечебных путевок, которыми пользовался Смирнов А.Ф. за последние 5 лет.

3. Определить количество членов профсоюза, которые пользовались туристическими путевками за последние 5 лет. Данные представить по годам в виде диаграммы.

**Вариант 4. Отдел кадров**

***Описание:*** Отдел кадров предприятия ведет учет сотрудников, работающих в подразделениях. Подразделения имеют название и номер.

Каждый сотрудник характеризуется ФИО, должностью, датой рождения, окладом, домашним адресом, служебным и домашним телефонами. Сотрудники имеют различное семейное положение и могут иметь детей. Дети характеризуются полом, именем, датой рождения.

***Задания:***

1. Составить список всех работников с указанием ФИО, наименования подразделения, домашнего адреса, возраст которых больше или равен 60 лет.

2. Составить список всех работников планового отдела с указанием ФИО и должности, имеющих детей младше 7 лет.

3. Определить средний возраст сотрудников каждого отдела.

4. Кто из сотрудников является самым молодым папой?

## **Вариант 5. Пенсионный фонд**

### ***Описание:***

Районный пенсионных фонд ведет учет денежных средств поступающих от предприятий района. Каждая организация представляет следующую информацию: Регистрационный номер в пенсионном фонд, ИНН, наименование, форму собственности, ФИО директора, ФИО главного бухгалтера. Дополнительно для каждого сотрудника предприятия указывается: № страхового свидетельства, ФИО сотрудника, месяц, год, сумма заработной платы.

Организации осуществляют перечисления в пенсионный фонд: № платежного поручения, дата, сумма, месяц, за который производится перечисление денег.

### ***Задания:***

1. Составить список организаций с указанием ИНН, наименования, ФИО директора, у которых перечисления в мае или июне 2002 года была меньше 10000 рублей.
2. Составить отчет о начисленной фонде заработной платы ОАО «Невинномысский Азот» за май 2003 года с указанием № страхового свидетельства, ФИО сотрудника, суммы заработной платы.
3. Определить в какой организации наибольшая средняя месячная зарплата сотрудников в 2003 г.

## **Вариант 6. Статистическое управление**

### ***Описание:***

Статистическое управление района хранит информацию о каждом предприятии – регистрационный номер, тип отрасли (сельское хозяйство, транспорт, легкая и тяжелая промышленность, строительство, материально-техническое снабжение), наименование, адрес, телефон, факс, форма собственности (государственно-федеральная, муниципально-городская, частная, акционерная), состав работающих: всего, мужчин, женщин, ИТР, рабочих, занимаемая площадь: общая площадь помещений, площадь участка, пло-

щадь участка под возможное строительство. Каждое предприятие может иметь несколько поставщиков сырья и комплектующих, (которые делятся на местных и внешних и имеют ИНН и наименование), может изготавливать несколько видов продукции с указанием его наименования и ежемесячного объема выпуска, может иметь несколько потребителей продукции (которые делятся на местных и внешних и имеют ИНН и наименование) на различные виды продукции.

***Задания:***

1. Составить отчет о внешних поставщиках предприятий, производящих металлообрабатывающие станки, чей ежемесячный объем выпуска данной продукции больше 10. В отчете указать ИНН, наименование организации поставщика.
2. Составить отчет о предприятиях, производящих школьную мебель и имеющих количество сотрудников больше 100 человек. В отчете указать ИНН, наименование организации, форму собственности.
3. Определить на каком из предприятий наибольший процент ИТР.

**Вариант 7. Библиотека**

***Описание:***

Библиографический отдел библиотеки ведет учет поступлений книг. Книга характеризуется одним или несколькими авторами (могут отсутствовать), шифром, названием, издательством, годом издания и количеством страниц. Книга может поступать в нескольких экземплярах и храниться в разных отделах библиотеки. Экземпляры имеют уникальные инвентарные номера.

Экземпляры книг могут выдаваться читателям. О каждом читателе хранится следующая информация: № читательского билета, № паспорта, ФИО, домашний адрес, место работы, возраст.

**Задания:**

1. Составить список книг, находящихся в читальном зале с указанием названия, автора, года издания.
2. Составить список читателей с указанием ФИО и возраста, которые брали Трудовое законодательство.
3. Составить список книг, которые брал Сидоров А.М. с указанием автора и названия книги.

**Вариант 8. Поликлиника****Описание:**

Регистратура ведет учет больных, посещающих поликлинику. Больной характеризуется № страхового полюса, ФИО, домашним адресом, полом и возрастом. Врач характеризуется ФИО, лечебным профилем (терапевт, хирург и т.п.) и имеет расписание (дата, время начала приема, время окончания приема, № кабинета). Больные назначаются к врачу соответствующего профиля по расписанию его работы.

**Задания:**

1. Составить список больных с указанием ФИО и возраста, назначенных на прием к терапевту Иванову С. П. 25.05.2002 г.
2. Составить список больных с указанием № страхового полюса, ФИО, домашнего адреса, назначенных на прием в 30 кабинет 25.05.2002 года с 9 часов до 14 часов. Дополнительно указать ФИО и лечебный профиль врача, принимавшего в это время в 30 кабинете.
3. Рассчитать средний возраст больных, посетивших хирурга в 2002 году.

**Вариант 9. Аптека****Описание:**

В базе данных хранятся сведения обо всех аптеках города, препаратах, продаваемых в аптеках и ценах на них. Аптека харак-



теризуется номером, названием, телефоном, адресом, ФИО директора. Товар характеризуется кодом, наименованием, страной производителем, фармацевтической группой (шифр, наименование). Товар может входить сразу в несколько фармацевтических групп. Один и тот же товар может продаваться разными аптеками по разной цене и по различным условиям оплаты (наличные, безналичные).

***Задания:***

1. Составить перечень аптек с указанием номера, названия и адреса, реализующих анальгетики по безналичному расчету.
2. Составить перечень товаров с указанием фармацевтической группы, произведенных в Бельгии и реализуемых аптекой №5.
3. Определить в какой аптеке анальгин продается по наиболее низкой цене.
4. Определить пять наиболее дорогих лекарств и где они производятся.

**Вариант 10. Научно-исследовательский институт**

***Описание:***

В НИИ производятся работы по различным темам. Каждая тема имеет уникальный шифр и наименование, руководителя темы (ФИО, должность, телефон), срок сдачи работ, общую сумму по договору. Одна и та же тема может выполняться несколькими подразделениями НИИ. Каждое подразделение характеризуется номером, наименованием, ФИО руководителя. Каждое подразделение может одновременно выполнять несколько тем.

***Задания:***

1. Составить список тем (шифр, наименование), которыми руководит Семенов Н.И. и указать какие подразделения выполняют эту тему.

2. Составить список руководителей подразделений (ФИО, шифр темы), выполняющих работы по темам, срок сдачи которых 01.06.2002 года.

3. Рассчитать сумму всех договоров, выполненных в течении 2002 года и какой процент эта сумма составляет от объема работ, выполненных в 2001 году.

### **Вариант 11. Заработная плата**

#### ***Описание:***

В бухгалтерии на основе данных о сотрудниках начисляется заработная плата. Имеются следующие сведения о сотрудниках: ФИО, наименование цеха, размер заработной платы за месяц, год рождения, дата поступления на работу, семейное положение, пол, количество детей, дата заболевания, дата выздоровления.

#### ***Задания:***

1. Составить ведомость на выдачу заработной платы. При начислении заработной платы учесть следующее: за время болезни начисляется 50%, если стаж работы менее 2 лет; 80%, если стаж работы от 2 до 4 лет и 100% при большем стаже. Удерживается 12% подоходный налог и 6% налог на бездетность.

2. Определить средний возраст рабочих в цехе и по заводу в целом.

3. Рассчитать средний заработок мужчин и женщин по цехам.

4. Определить фамилии сотрудников по цехам, чей заработок меньше N.

**МЕТОДЫ И СРЕДСТВА ПРОЕКТИРОВАНИЯ  
ИНФОРМАЦИОННЫХ СИСТЕМ И  
ТЕХНОЛОГИЙ**

*методические указания к выполнению лабораторных работ  
для направления 09.03.02 –  
Информационные системы и технологии*

Составитель *Д.В. Болдырев*

---

Редактор

---

Подписано в печать \_\_. \_\_. \_\_      Формат 60×84 1/16  
Уч.-изд. л. 2,25    Усл. печ. л. 2,3    Тираж 50 экз.    Заказ №  
Невинномысский технологический институт (филиал)  
ФГАОУ ВПО «Северо-Кавказский федеральный университет»

---

Типография \_\_\_\_\_