

**Министерство образования и науки Украины
Донбасская государственная машиностроительная академия**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к компьютерному практикуму

по дисциплине

«ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ СЛОЖНЫХ СИСТЕМ»

(для студентов заочной формы обучения
специальности 151)

Утверждено
на заседании
методического совета
Протокол № от

Краматорск 2018

Методические указания к компьютерному практикуму по дисциплине "Технология программирования сложных систем". (для студентов заочной формы обучения специальности 151)

Приведено описание интерфейса инструментального средства визуального моделирования и автоматической разработки программного обеспечения Rational Rose. Изложены приемы визуализации, определения, документирования и конструирования компонентов программной системы. Даны методические рекомендации по разработке моделей.

1 ИЗУЧЕНИЕ РАБОЧЕГО ИНТЕРФЕЙСА CASE-СРЕДСТВА RATIONAL ROSE

Цель лабораторной работы: освоение интерфейса и приемов работы с программным продуктом **Rational Rose**.

1.1 Общая характеристика CASE-средства IBM Rational Rose 2003

В настоящее время **CASE-средство IBM Rational Rose 2003** (release 2003.06.00) представляет собой современный интегрированный инструментарий для анализа, моделирования, проектирования архитектуры и разработки программных систем.

В рамках общего продукта IBM Rational Rose существуют различные варианты этого средства, отличающиеся между собой диапазоном предоставляемых возможностей. Базовым средством в настоящее время является **IBM Rational Rose Enterprise Edition**, которое обладает наиболее полными возможностями.

Функциональные особенности этой программы заключаются в следующем:

- интеграция с **MS Visual Studio 6**, которая включает поддержку на уровне прямой и обратной генерации кодов и диаграмм **Visual Basic** и **Visual C++** с использованием **Microsoft ATL (Active Template Library)**, **Web-Классов**, **DHTML** и протоколов доступа к различным базам данных;
- непосредственная работа (инжиниринг и реинжиниринг) с исполняемыми модулями и библиотеками форматов **EXE**, **DLL**, **TLB**, **OCX**.
- поддержка технологий **MTS (Microsoft Transaction Server)** и **ADO (ActiveX Data Objects)** на уровне шаблонов и исходного кода, а также элементов технологии Microsoft – **COM+ (DCOM)**;
- полная поддержка компонентов **CORBA** и **J2EE**, включая реализацию технологии компонентной разработки приложений **CBD (Component-Based Development)**, языка определения интерфейса **IDL (Interface Definition Language)** и языка определения данных **DDL (Data Definition Language)**;
- полная поддержка среды разработки Java-приложений, включая прямую и обратную генерацию классов **Java** формата **JAR**, а также работу с файлами формата **CAB** и **ZIP**.

1.2 Особенности рабочего интерфейса программы Rational Rose 2003

В **CASE-средстве IBM Rational Rose 2003** реализованы общепринятые стандарты на рабочий интерфейс программы, аналогично известным средам визуального программирования. Рабочий интерфейс программы **Rational**

Rose (рис. 1.1) состоит из различных элементов, основными из которых являются:

- главное меню;
- стандартная панель инструментов;
- специальная панель инструментов;
- окно браузера (браузера) проекта;
- рабочая область изображения диаграммы или окно диаграммы;
- окно документации;
- окно журнала.

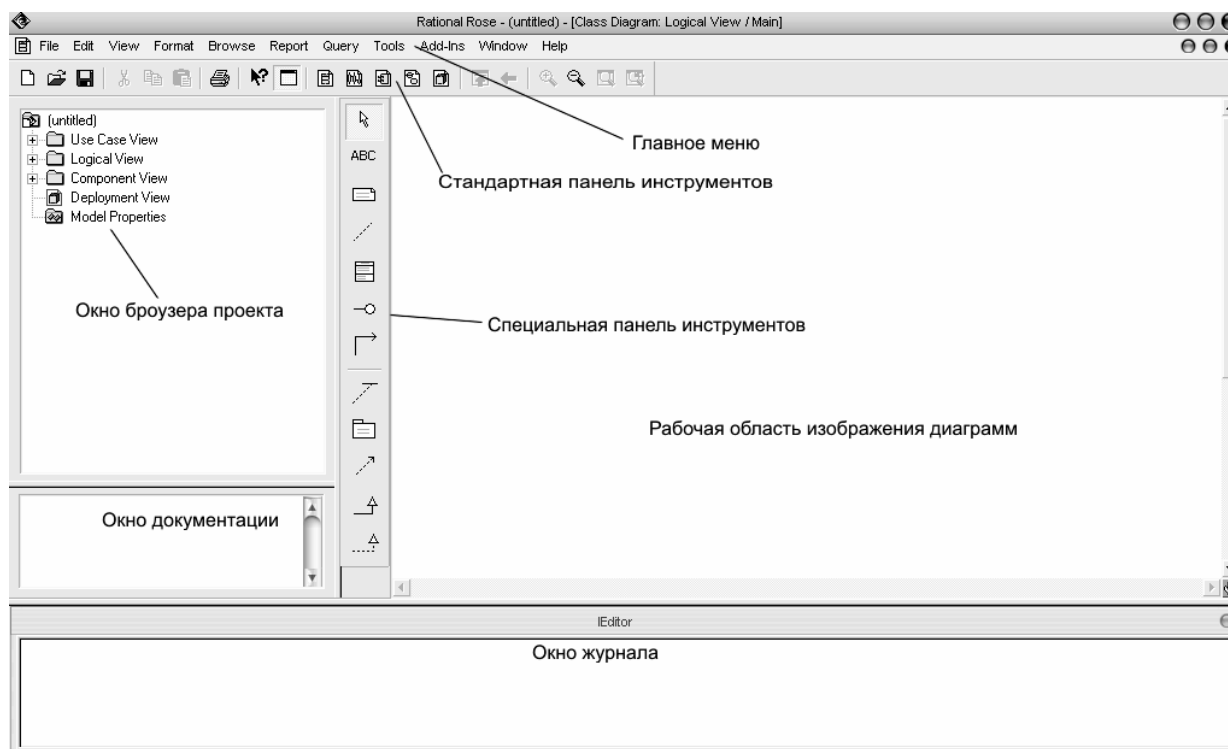


Рисунок 1.1 – Общий вид рабочего интерфейса CASE-средства IBM Rational Rose 2003

1.3 Главное меню и стандартная панель инструментов

Главное меню программы **IBM Rational Rose 2003** выполнено в общепринятом стандарте и имеет следующий вид (рис. 1.2).

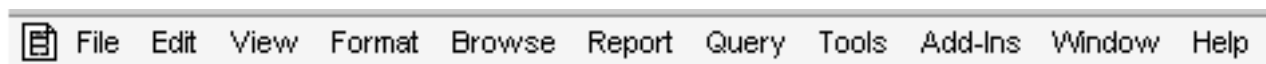


Рисунок 1.2 – Внешний вид главного меню программы

Отдельные пункты меню объединяют сходные операции, относящиеся ко всему проекту в целом. Некоторые из пунктов меню содержат хорошо

знакомые операции, такие как открытие проекта, вывод на печать диаграмм, копирование в буфер и вставка из буфера различных элементов диаграмм. Другие операции специфичны, что может потребовать дополнительных усилий для их изучения (свойства операций генерации программного кода или проверки согласованности моделей).

Стандартная панель инструментов располагается ниже строки главного меню и имеет вид, показанный на рисунке 1.3. Стандартная панель инструментов обеспечивает быстрый доступ к тем командам меню, которые выполняются разработчиками наиболее часто.

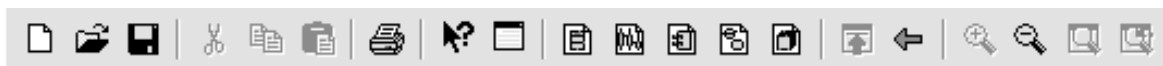


Рисунок 1.3 – Внешний вид стандартной панели инструментов

Пользователь может настроить внешний вид этой панели по своему усмотрению. Для этого необходимо выполнить операцию главного меню: **Tools ► Options** (Инструменты ► Параметры), открыть в появившемся диалоговом окне вкладку **Toolbars** (Панели инструментов) и нажать кнопку **Standard** (Стандартная). В дополнительно открытом окне можно переносить требуемые кнопки из левого списка в правый список, а ненужные кнопки – из правого списка в левый. Данным способом можно показать или скрыть различные кнопки инструментов, а также изменить их размер. Назначение отдельных кнопок стандартной панели инструментов приводится далее при рассмотрении операций главного меню.

1.4 Назначение операций главного меню File и Edit

Рабочий интерфейс средства **IBM Rational Rose 2003** имеет главное меню, которое позволяет пользователю загружать и сохранять информацию во внешних файлах, изменять внешний вид элементов графического интерфейса, вызывать справочную информацию, а также другие диалоговые окна для работы с программой **IBM Rational Rose 2003**.

Операции главного меню **File** (Файл) позволяют создавать новые модели в нотации языка UML, загружать и сохранять разрабатываемую модель во внешнем файле, распечатывать на принтере разработанные диаграммы. Назначение операций этого пункта главного меню представлено в таблице 1.1.

Операции главного меню **Edit** (Редактирование) позволяют выполнять действия по редактированию элементов модели и их свойств, а также выполнять поиск элементов в рамках разрабатываемого проекта. Назначение операций этого пункта главного меню представлено в таблице 1.2.

Таблица 1.1 – Операции пункта главного меню **File** (Файл)

Название операции меню	Назначение операции
New	Создает новую модель с именем untitled (по умолчанию)
Open	Вызывает стандартное диалоговое окно открытия внешнего файла с диска. При этом открыть можно либо файл модели (расширение .mdl), либо файл подмодели (расширение .ptl)
Save	Сохраняет модель во внешнем файле на диске
Save As	Сохраняет модель под другим именем
Save Log As	Сохраняет содержимое журнала ошибок во внешнем файле на диске с именем error.log
AutoSave log	Автоматически сохраняет содержимое журнала ошибок во внешнем файле с именем error.log
Clear Log	Очищает содержимое журнала
Load Model Workspace	Загружает рабочую область из внешнего файла на диске. При этом файл должен иметь расширение .wsp
Save Model Workspace	Сохраняет рабочую область модели во внешнем файле на диске с расширением .wsp
Save Model Workspace As	Сохраняет рабочую область модели во внешнем файле на диске с расширением .wsp под другим именем
Units	Загружает категорию элементов модели из внешнего файла на диске. При этом открыть можно файл с расширением .cat
Import	Позволяет импортировать информацию из внешних файлов различных форматов
Export Model	Позволяет экспортировать информацию о модели во внешний файл
Update	Позволяет вставить информацию обратного проектирования из внешнего файла с расширением .red в разрабатываемую модель
Print	Позволяет распечатать отдельные диаграммы и спецификации модели путем их выбора из диалогового окна
Print Setup	Позволяет настроить свойства печати
Edit Path Map	Вызывает окно задания путей доступа к файлам Rational Rose

Таблица 1.2 – Операции пункта главного меню **Edit** (Редактирование)

Название операции меню	Назначение операции
Undo	Отменяет выполнение последнего действия по удалению или перемещению элементов модели
Redo	Восстанавливает изображение диаграммы после отмены операции перемещения
Cut	Вырезает выделенный элемент модели и помещает его в буфер
Copy	Копирует выделенный элемент модели и помещает его в буфер
Paste	Вставляет элемент из буфера обмена
Delete	Удаляет выделенные элементы из текущей диаграммы, но не из модели
Select All	Выделяет все элементы на текущей диаграмме
Delete from Model	Удаляет все выделенные элементы из модели
Relocate	Позволяет перемещать или отменять перемещение классов, ассоциаций или компонентов из одного пакета в другой
Find	Вызывает меню поиска элемента
Reassign	Позволяет заменить выделенный элемент другим элементом модели
Compartment	Позволяет отображать дополнительную информацию об объектах, классах, актерах или пакетах
Change Info	Позволяет изменить тип выделенного элемента диаграммы

1.5 Назначение операций главного меню **View**, **Format** и **Browse**

Операции главного меню **View** (Вид) позволяют отображать на экране различные элементы рабочего интерфейса и изменять графическое представление диаграмм. Назначение операций этого пункта главного меню представлено в следующей таблице (табл. 1.3).

Таблица 1.3 – Операции пункта главного меню *View* (Вид)

Название операции меню	Назначение операции главного меню
Toolbars	Позволяет настроить внешний вид рабочего интерфейса системы IBM Rational Rose 2003 и содержит дополнительные подпункты: Standard – делает видимой/невидимой стандартную панель инструментов (рис. 1.3); Toolbox – делает видимой/невидимой специальную панель инструментов текущей активной диаграммы; Configure – вызывает диалоговое окно настройки параметров модели, открытое на вкладке настройки панелей инструментов
Status Bar	Делает видимой/невидимой строку состояния
Documentation	Делает видимым/невидимым окно документации
Browser	Делает видимым/невидимым браузер проекта
Log	Делает видимым/невидимым окно журнала
Editor	Делает видимым/невидимым встроенный текстовый редактор
Time Stamp	Включает/выключает режим отображения времени в записях журнала
Zoom to Selection	Изменяет масштаб изображения выделенных элементов модели, так чтобы они разместились в одном окне
Zoom In	Увеличивает масштаб изображения
Zoom Out	Уменьшает масштаб изображения
Fit in Window	Изменяет масштаб изображения всех элементов текущей диаграммы, так чтобы все они разместились в одном окне
Undo Fit in Window	Отменяет изменение масштаба изображения для размещения элементов в одном окне
Page Breaks	Разбивает текущую диаграмму на страницы для последующей печати
Refresh	Перерисовывает текущую диаграмму
As Booch	Изображает элементы в нотации Г. Буча
AsOMT	Изображает элементы в нотации OMT
As Unified	Изображает элементы в нотации UML

Операции главного меню **Format** (Формат) позволяют выполнять действия по изменению внешнего вида элементов модели на различных диаграммах. Назначение операций этого пункта главного меню представлено в таблице 1.4.

Таблица 1.4 – Операции пункта главного меню **Format** (Формат)

Название операции меню	Назначение операции главного меню
Font Size	Изменяет масштаб используемого шрифта
Font	Вызывает диалоговое окно выбора шрифта
Line Color	Вызывает диалоговое окно выбора цвета линий
Fill Color	Вызывает диалоговое окно выбора цвета для изображения графических элементов диаграмм
Use Fill Color	Включает/выключает режим отображения цвета для изображения графических элементов диаграмм
Automatic Resize	Включает/выключает режим автоматического изменения размеров графических элементов диаграмм для отображения текстовой информации об их свойствах
Stereotype	Позволяет выбрать способ изображения стереотипов выделенных элементов и содержит подпункты: None – стереотип не показывается; Label – стереотип отображается в форме текста; Decoration – стереотип отображается в форме небольшой пиктограммы в правом верхнем углу графического элемента; Icon – элемент диаграммы отображается в форме специального графического стереотипа
Stereotype Label	Включает/выключает режим отображения текстовых стереотипов (ассоциаций, зависимостей и пр.) диаграммы
Show Visibility	Включает/выключает режим отображения кванторов видимости
Show Compartment Stereotypes	Включает/выключает режим отображения текстовых стереотипов атрибутов и операций выделенных классов
Show Operation Signature	Включает/выключает режим отображения сигнатуры операций выделенных классов
Show All Attributes	Делает видимыми/невидимыми атрибуты выделенных классов
Show All Operations	Делает видимыми/невидимыми операции выделенных классов

Продолжение табл. 1.4

Название операции меню	Назначение операции главного меню
Suppress Attributes	Делает видимой/невидимой секцию атрибутов выделенных классов. Скрывает секцию атрибутов даже в том случае, когда выбрана опция Show All Attributes
Suppress Operations	Делает видимой/невидимой секцию операций выделенных классов. Скрывает секцию операций даже в том случае, когда выбрана опция Show All Operations
Line Style	Позволяет выбрать способ графического изображения линий взаимосвязей и содержит дополнительные подпункты: Rectilinear – линия изображается в форме вертикальных и горизонтальных отрезков; Oblique – линия изображается в форме наклонных отрезков; Toggle – промежуточный вариант изображения линии
Layout Diagram	Позволяет автоматически разместить графические элементы в окне диаграммы с минимальным количеством пересечений и наложений соединительных линий
Autosize All	Позволяет автоматически изменить размеры графических элементов текущей диаграммы таким образом, чтобы текстовая информация помещалась внутри изображений соответствующих элементов
Layout Selected Shapes	Позволяет автоматически разместить выделенные графические элементы в окне диаграммы с минимальным количеством пересечений и наложений соединительных линий

Операции главного меню **Browse** (Обзор) позволяют отображать рабочие окна с различными каноническими диаграммами разрабатываемой модели и вызывать диалоговые окна редактирования свойств отдельных элементов модели.

Операции этого пункта главного меню приведены в таблице 1.5.

Таблица 1.5 – Назначение операций пункта меню **Browse** (Обзор)

Название операции меню	Назначение операции
Use Case Diagram Class Diagram Component Diagram Deployment Diagram Interaction Diagram State Machine Diagram	Позволяет отобразить в рабочем окне соответствующую диаграмму разрабатываемой модели
Expand	Отображает в рабочем окне первую из диаграмм выделенного пакета модели
Parent	Отображает в рабочем окне родителя выделенной диаграммы модели
Specification	Вызывает диалоговое окно свойств выделенного элемента модели
Top Level	Отображает в рабочем окне диаграмму самого верхнего уровня для текущей диаграммы модели
Referenced Item	Отображает в рабочем окне диаграмму классов, содержащую класс для выделенного объекта модели
Previous Diagram	Отображает в рабочем окне предыдущую диаграмму модели
Create Message Trace Diagram	Позволяет создать диаграмму трассировки сообщений
Top Level	Отображает в рабочем окне диаграмму самого верхнего уровня для текущей диаграммы модели
Referenced Item	Отображает в рабочем окне диаграмму классов, содержащую класс для выделенного объекта модели
Previous Diagram	Отображает в рабочем окне предыдущую диаграмму модели
Create Message Trace Diagram	Позволяет создать диаграмму трассировки сообщений

1.6 Окно браузера проекта

Окно браузера проекта по умолчанию располагается в левой части рабочего интерфейса ниже стандартной панели инструментов и имеет вид, показанный на рисунке 1.4.

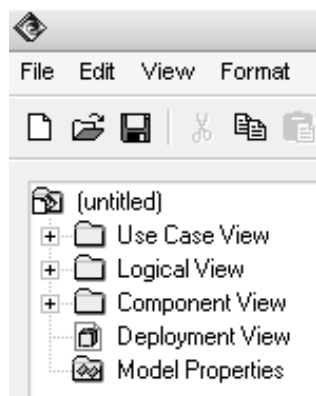


Рисунок 1.4 – Внешний вид браузера проекта с иерархическим представлением его структуры

Броузер проекта организует представления модели в виде иерархической структуры, которая упрощает навигацию и позволяет отыскать любой элемент модели в проекте. При этом самая верхняя строка браузера проекта содержит имя разрабатываемого проекта (по умолчанию – **untitled**). Любой элемент, который разработчик добавляет в модель, сразу отображается в окне браузера. Соответственно, выбрав элемент в окне браузера, мы можем его визуализировать в окне диаграммы или изменить его спецификацию.

Броузер проекта позволяет также организовывать элементы модели в пакеты и перемещать элементы между различными представлениями модели. При желании окно браузера можно расположить в другом месте рабочего интерфейса либо скрыть вовсе, используя для этого операцию **Browser** главного меню **View** (Вид). Можно также изменить размеры браузера.

Иерархическое представление структуры каждого разрабатываемого проекта организовано в форме 4-х представлений:

- **Use Case View** – представление вариантов использования, в котором содержатся диаграммы вариантов использования и их реализации в виде вариантов взаимодействия;
- **Logical View** – логическое представление, в котором содержатся диаграммы классов, диаграммы состояний и диаграммы деятельности;
- **Component View** – представление компонентов, в котором содержатся диаграммы компонентов разрабатываемой модели;

- **Deployment View** – представление развертывания, в котором содержится единственная диаграмма развертывания разрабатываемой модели.

При создании нового проекта иерархическая структура формируется программой автоматически.

1.7 Специальная панель инструментов и окно диаграммы

Специальная панель инструментов располагается между окном браузера и окном диаграммы в средней части рабочего интерфейса. По умолчанию предлагается панель инструментов для построения диаграммы классов (рис. 1.5).



Рисунок 1.5 – Внешний вид специальной панели инструментов для диаграммы классов

Расположение специальной панели инструментов можно изменять, переместив рамку панели в нужное место. Программа Rational Rose позволяет настраивать состав кнопок данной панели, добавляя или удаляя отдельные кнопки, соответствующие тем или иным инструментам. Названия кнопок данной панели всегда можно узнать из всплывающих подсказок, появляющихся после задержки указателя мыши над соответствующей кнопкой.

Внешний вид специальной панели инструментов зависит не только от выбора типа разрабатываемой диаграммы, но и от выбора графической нотации для изображения самих элементов этих диаграмм. В Rational Rose 2003 реализованы три таких нотации: UML, OMT и Booch (последние две нотации практически не используются на практике).

Окно диаграммы является основной графической областью программы Rational Rose, в которой визуализируются различные представления модели проекта. По умолчанию окно диаграммы располагается в правой части рабочего интерфейса, однако его расположение и размеры также можно

изменить. Если не был использован мастер проектов, окно диаграммы представляет собой чистую область, не содержащую никаких элементов модели (рис. 1.1). При разработке диаграмм в окне диаграммы будут располагаться соответствующие графические элементы модели (рис. 1.6).

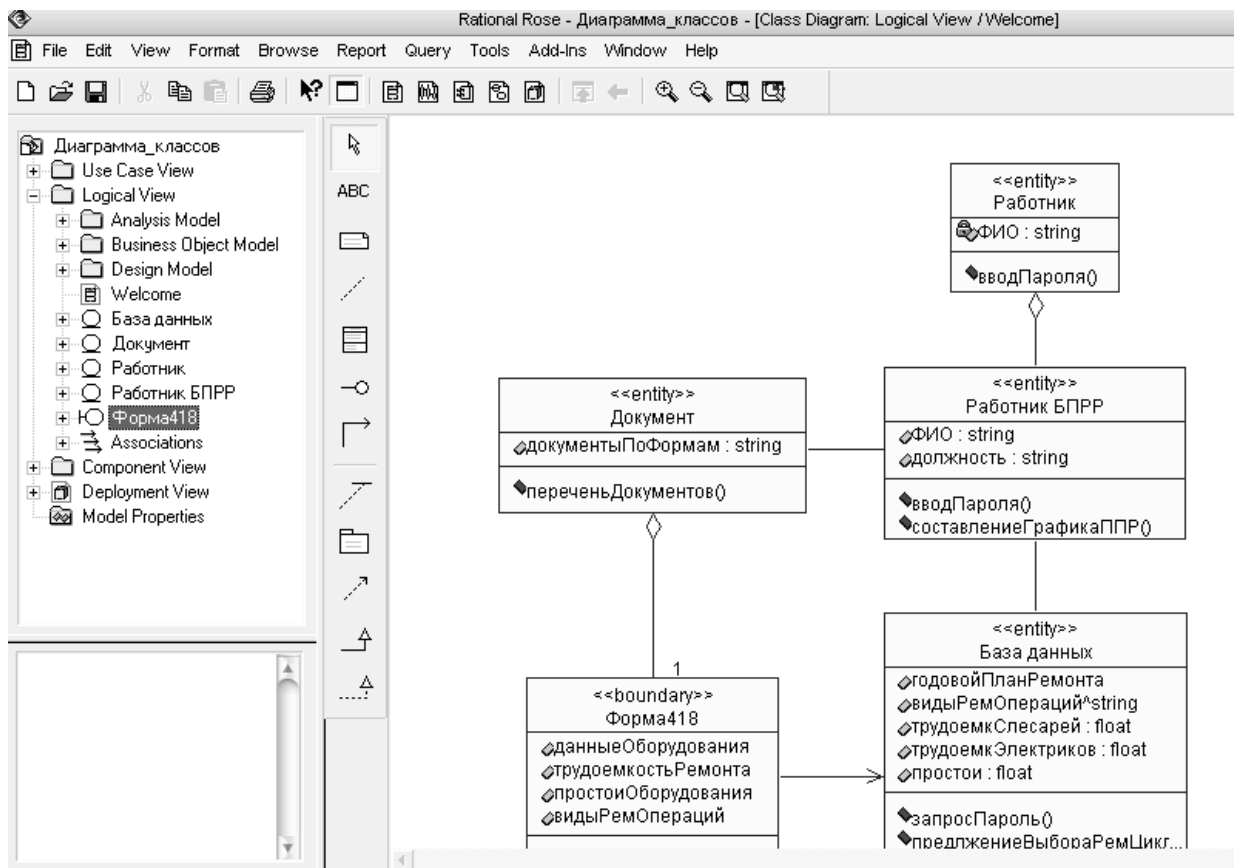


Рисунок 1.6 – Внешний вид окна диаграммы с диаграммой классов модели

Название диаграммы, которая является активной и располагается в данном окне, развернутом на всю область диаграммы, указывается в строке заголовка программы Rational Rose. Если же окно диаграммы не развернуто на всю область диаграммы, то название диаграммы указывается в строке заголовка окна диаграммы.

Одновременно в графической области диаграмм могут присутствовать несколько окон диаграмм; при этом активной может быть только одна из них.

Переключение между диаграммами можно осуществить выбором нужного представления на стандартной панели инструментов, а также с помощью выделения требуемой диаграммы в браузере проекта или с помощью операций главного меню **Window** (Окно). При активизации отдельного вида диаграммы изменяется внешний вид специальной панели инструментов, которая настраивается под конкретный вид диаграммы.

1.8 Окно документации и окно журнала

Окно документации по умолчанию должно присутствовать на экране после загрузки программы. Если по какой-то причине оно отсутствует, то его можно отобразить через пункт меню **View ► Documentation** (Вид ► Документация), после чего окно документации появится ниже окна браузера проекта (рис. 1.7). Окно документации, как следует из его названия, предназначено для документирования элементов разрабатываемой модели. В него можно записывать различную текстовую информацию, и что важно – на русском языке. Эта информация при генерации программного кода преобразуется в комментарии и никак не влияет на логику выполнения программного кода.

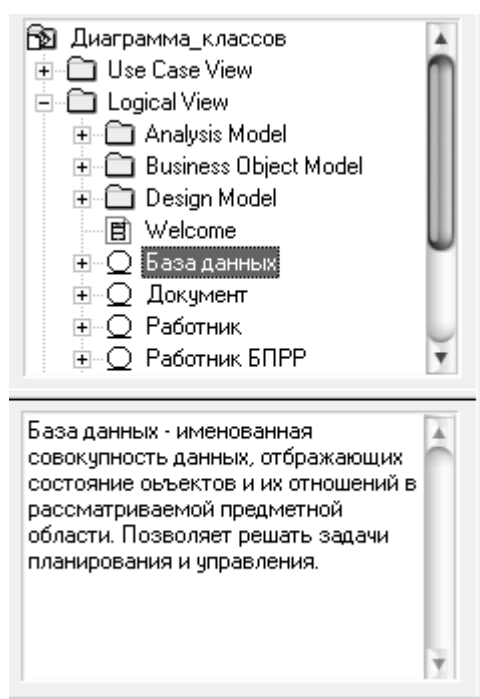


Рисунок 1.7 – Внешний вид окна документации (внизу) с информацией о классе “База данных”

В окне документации активизируется та информация, которая относится к выделенному элементу диаграммы или к диаграмме в целом. Выделить элемент можно либо в окне браузера, либо в окне диаграммы. При добавлении нового элемента на диаграмму, например, класса, документация к нему является пустой (**No documentation**). Разработчик самостоятельно вносит необходимую пояснительную информацию, которая запоминается программой и может быть изменена в ходе работы над проектом. Размеры и положение окна документации можно изменять по своему усмотрению.

Окно журнала (**Log**) предназначено для автоматической записи различной служебной информации в ходе работы с программой. В журнале фиксируется время и характер выполняемых разработчиком действий, таких

как обновление модели, настройка меню и панелей инструментов, а также сообщений об ошибках, возникающих при генерации программного кода. Окно журнала изображается поверх других окон в нижней области рабочего интерфейса программы (рис. 1.1).

Если окно журнала отсутствует на экране, то отобразить его можно с помощью операции главного меню **View ► Log** (Вид ► Журнал), для чего следует выставить отметку в соответствующей строке вложенного меню для данной операции. С целью увеличения размеров графической области диаграммы окно журнала чаще всего убирают с экрана, что можно выполнить с помощью кнопки закрытия этого окна в верхнем левом его углу или убрав отметку в соответствующей строке вложенного меню **View ► Log**.

1.9 Назначение операций главного меню **Report, Query** и **Tools**

Операции главного меню **Report** (Отчет) позволяют отображать различную информацию об элементах разрабатываемой модели и вызывать диалоговое окно выбора шаблона для генерации отчета о модели. Назначение операций этого пункта главного меню представлено в таблице 1.6.

Таблица 1.6 – Операции пункта главного меню **Report** (Отчет)

Название операции меню	Назначение операции главного меню
Show Usage	Отображает в диалоговом окне информацию об использовании выделенного элемента модели на различных
Show Instances	Отображает в диалоговом окне информацию об использовании объектов выделенного класса модели на
Show Access Violations	Отображает в диалоговом окне информацию о ссылках классов одного пакета на классы другого пакета при отсутствии соответствующей зависимости доступа или импорта между этими пакетами в модели
SoDA Report	Позволяет сгенерировать отчет о разрабатываемой модели в формате MS Word с использованием специального средства IBM Rational SoDA
Show Participants in UC	Отображает в диалоговом окне информацию о классах, компонентах и операциях, которые участвуют в реализации выделенного варианта использования модели на различных диаграммах

Операции главного меню **Query** (Запрос) позволяют добавлять существующие элементы разрабатываемой модели на редактируемую диаграмму. Назначение операций этого пункта главного меню представлено в таблице 1.7.

Таблица 1.7 – Операции пункта главного меню **Query** (Запрос)

Название	Назначение операции главного меню
Add Classes	Вызывает диалоговое окно для добавления на текущую диаграмму классов
Add Use Cases	Вызывает диалоговое окно для добавления на текущую диаграмму вариантов использования
Expand Selected Elements	Вызывает диалоговое окно для добавления на текущую диаграмму элементов модели
Hide Selected Elements	Вызывает диалоговое окно с предложением удалить элементы модели, связанные с выделенным элементом
Filter Relationships	Вызывает диалоговое окно, позволяющее включить/выключить режим отображения

Состав операций пункта главного меню **Tools** (Инструменты) зависит от установленных в программе Rational Rose конкретных расширений. Назначение операций этого пункта главного меню для типовой конфигурации программы представлено в таблице 1.8.

Таблица 1.8 – Операции пункта главного меню **Tools** (Инструменты) типовой конфигурации

Название операции	Назначение операции главного меню
Model Properties	Позволяет выполнить настройку свойств языка реализации для выделенного элемента модели и содержит подпункты: Edit – редактирование набора свойств; View – просмотр набора свойств; Replace – замена существующего набора свойств на новый, загружаемый из внешнего файла с расширением .prp или .pty ; Export – сохранение существующего набора свойств во внешнем файле с расширением .prp или .pty ; Add – добавление к существующему набору свойств нового набора свойств, загружаемого из внешнего файла с расширением .prp или .pty ; Update – обновление существующего набора свойств после его редактирования или дополнения

Продолжение табл. 1.8

Название операции	Назначение операции главного меню
Check Model	Проверяет разрабатываемую модель на наличие ошибок, информация о которых отображается в окне журнала
Create	Создает новый элемент модели из предлагаемого списка
Options	Вызывает диалоговое окно настройки параметров модели, открытое на вкладке General
Open Script	Вызывает стандартное диалоговое окно открытия внешнего файла, содержащего текст скрипта (файл с расширением . ebs) для его редактирования в окне встроенного редактора скриптов
New Script	Открывает дополнительное окно встроенного редактора скриптов для создания, отладки, выполнения и сохранения нового скрипта во внешнем файле с расширением . ebs
ANSI C++	Позволяет выполнить настройку свойств языка программирования ANSI C++ , выбранного в качестве языка реализации отдельных элементов модели
CORBA	Позволяет выполнить настройку свойств и спецификацию модели для генерации объектов CORBA для реализации отдельных элементов модели
Java/J2EE	Позволяет выполнить настройку свойств языка программирования Java , выбранного в качестве языка реализации отдельных элементов модели
Oracle8	Позволяет выполнить настройку свойств и спецификацию модели для генерации схем СУБД Oracle 8 для отдельных элементов модели
Quality Architect	Позволяет выполнить настройку свойств и тестирование модели
Rational Requisite Pro	Позволяет выполнить настройку свойств модели для установления связей со специальным средством спецификации и управления требованиями
Model Integrator	Открывает рабочее окно специального средства интеграции моделей
Web Publisher	Позволяет выполнить настройку свойств модели для ее публикации в гипертекстовом формате
TOPLink	Вызывает мастер преобразования таблиц модели данных в классы языка программирования Java , выбранного в качестве языка реализации отдельных элементов модели
COM	Позволяет выполнить настройку свойств и спецификацию модели для генерации объектов COM с целью реализации отдельных элементов модели

Продолжение табл. 1.8

Название операции	Назначение операции главного меню
Visual C++	Позволяет выполнить настройку свойств и спецификацию модели для генерации программного кода MS Visual C++ , выбранного в качестве языка реализации отдельных элементов модели
Version Control	Позволяет выполнить настройку свойств модели для установления связей со специальным средством управления и контроля версий модели
Visual Basic	Позволяет выполнить настройку свойств и спецификацию модели для генерации программного кода MS Visual Basic , выбранного в качестве языка реализации отдельных элементов модели
XML_DTD	Позволяет выполнить настройку свойств и спецификацию модели для ее публикации в формате расширяемого языка разметки XML
Class Wizard	Вызывает мастер создания нового класса и его размещения на выбранной диаграмме модели

1.10 Сведения об отчете

Отчет по данной работе не составляется.

Освоение и защита работы оцениваются по результатам тестирования. Содержание тестов определяется требованиями *знаний особенностей* применения программы Rational Rose и *умений использовать* это CASE-средство для моделирования программ и создания программного кода.

2 РАЗРАБОТКА ДИАГРАММ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

Цель лабораторной работы: освоение приемов и методики разработки диаграмм вариантов использования с применением программы Rational Rose.

2.1 Рекомендации по разработке вариантов использования

Наиболее важный вопрос, который следует задать, приступая к работе над новой программной системой, звучит так: **"Какая именно система должна быть создана и зачем?"**. Процесс формулирования целей будущей системы наряду с обдумыванием формы ее представления и общих ограничений составляет начальную фазу жизненного цикла проекта, которая завершается заявлением следующего вида: **"Система должна выполнять..."**.

На начальной фазе разработки формулируются основные задачи и рассматриваются многочисленные исходные предположения, одни из которых принимаются, а другие отвергаются. Результатом правильного выполнения начальной фазы служат достаточно подробные требования к технологическим качествам будущей системы и организационным условиям ее применения.

Характеристики поведения разрабатываемой системы фиксируются и документируются средствами модели, которая отображает функции продукта (*варианты использования – use case*), представляет окружение системы (множество *активных субъектов – actors*) и определяет связи между вариантами использования и активными субъектами (*диаграммы вариантов использования – use case diagrams*).

Выбор вариантов использования. Варианты использования (*use case*) позволяют моделировать диалог между активным субъектом и системой, отображая предоставляемые системой функции. Набор вариантов использования – это последовательность выполняемых системой транзакций, которая приводит к желаемому для активного субъекта результату.

При выборе вариантов использования системы необходимо сформулировать ответы на следующие вопросы.

1. Какие активные субъекты участвуют в работе системы, и какие задачи решает каждый из субъектов?
2. Какие данные в контексте системы тот или иной субъект может создавать, сохранять, изменять, удалять или считывать?
3. Какие варианты использования гарантируют обработку данных?
4. Должны ли активные субъекты сообщать системе о каких-либо непредвиденных внешних обстоятельствах?
5. Имеет ли субъект право получать информацию об определенных

событиях, происходящих в системе?

6. Какие варианты использования связаны с поддержкой и администрированием системы?
7. Удовлетворяются ли вариантами использования все функциональные требования, предъявляемые к системе?

Выбор активных субъектов. Каждый из внешних *активных субъектов (actors)* отождествляется с чем-то или с кем-то, взаимодействующим с системой. Активный субъект способен выполнять различные функции:

- только вводить данные в систему;
- только получать информацию из системы;
- взаимодействовать с системой в обоих направлениях.

Помощь в отыскании активных субъектов могут дать ответы на следующие вопросы.

- Кто именно заинтересован в выполнении определенного требования?
- В каком подразделении организации должна использоваться система?
- Кто получит преимущества от внедрения системы в эксплуатацию?
- Кто будет поставлять системе те или иные данные, обращаться к ним и нести ответственность за их обновление и удаление?
- Кому предстоит выполнять обязанности администратора системы?
- Должна ли система использовать внешние ресурсы?
- Способен ли один и тот же активный субъект играть несколько различных ролей?
- Разрешено ли нескольким субъектам осуществлять в системе одинаковые функции?
- Будет ли система использоваться совместно с какими-либо существующими унаследованными системами?

Процедура определения активных субъектов системы отличается итеративным характером – первый вариант списка субъектов редко бывает окончательным. Внимательно анализируя роли сущностей, взаимодействующих с системой, можно прийти к адекватному множеству активных субъектов.

Формирование потоков событий. С каждым вариантом использования связан определенный поток событий, происходящих по мере выполнения соответствующих функций системы. При описании потока событий определяется, *что* необходимо осуществить, и игнорируются аспекты того, *как* это делается. Другими словами, события воспроизводятся средствами языка предметной области, а не в терминах практической

реализации функций. Схема потока событий должна содержать ответы на следующие вопросы.

- Как и когда вариант использования инициируется и завершается?
- Каким образом активные субъекты взаимодействуют с системой?
- Какие данные затрагиваются вариантом использования?
- Что представляет собой нормальная последовательность событий, предусмотряваемых вариантом использования?
- Существуют ли альтернативные потоки событий?

Документирование потока событий. Потоки событий обычно документируются по мере потребности. Описание потока событий для варианта использования системы содержится в документе **Use Case Specification** (спецификация варианта использования). Для создания подобного документа в каждом проекте должен применяться некий стандартный шаблон. Рекомендуется применение шаблона из регламента **Rational Unified Process**, в котором предусмотрена следующая информация:

- 1.0. Наименование варианта использования.
 - 1.1. Краткое описание.
- 2.0. Потоки событий.
 - 2.1. Основной поток.
 - 2.2. Альтернативные потоки.
 - 2.2.*. <Альтернативный поток *>.
- 3.0. Специальные требования
 - 3.*. <Специальное требование *>.
- 4.0. Предусловия.
 - 4.*. <Предусловие *>.
- 5.0. Постусловия.
 - 5.* <Постусловие *>.
- 6.0. Дополнительные замечания.
 - 6.*. Дополнительное замечание *.

Назначение связей в вариантах использования. Между активным субъектом и вариантом использования системы допускается устанавливать *связь ассоциации (association relationship)*, которая выполняет коммуникативную функцию, сообщая о взаимодействии субъекта с системой в рамках определенного варианта использования. Направление связи указывает, кто (субъект или система) является инициатором взаимодействия. Двухнаправленная ассоциация представляется в модели в виде линии, соединяющей связываемые элементы. Возможность взаимодействия элементов только в одном направлении отображается с помощью стрелки.

Варианты использования могут соединяться *связями зависимости (dependency relationships)* двух типов – *включения (include)* и *расширения (extend)*.

Включающей связью соединяются определенный вариант использования и любой другой вариант, который предполагает выполнение функций, дополняющих вариант использования. Включающая связь отображается в модели с помощью стрелки, направленной от базового варианта к включаемому.

Расширяющие связи применяются для описания:

- множеств необязательных функций;
- поведения системы при возникновении нештатных ситуаций;
- различных потоков событий, инициируемых в зависимости от того, какие опции выбираются пользователем.

Примером может служить вариант использования системы управления конвейером, адресующий специальный "аварийный" вариант при возникновении технических неполадок. Расширяющая связь отображается в модели с помощью стрелки, направленной к базовому варианту от расширяющего варианта.

В UML поддерживается понятие *стереотипа (stereotype)*, которое обеспечивает возможность создания новых элементов модели на основе базовых и позволяет определять минимальное множество символов, пополняемое с целью описания набора сущностей. Имена стереотипов заключаются в угловые кавычки <<...>> и размещаются рядом со стрелками, описывающими связи.

Ассоциация может быть снабжена меткой с именем стереотипа <<communicate>>, подчеркивающей, что связь обладает коммуникативной функцией, хотя это не обязательно, поскольку ассоциация – единственный тип связей, которыми могут соединяться активные субъекты и варианты использования. Включающие и расширяющие связи, напротив, относятся к одной и той же категории связей – *зависимости* и поэтому *должны* снабжаться соответствующими метками. Пример диаграммы, содержащей ассоциацию и связи зависимости, приведен на рисунке 2.1.

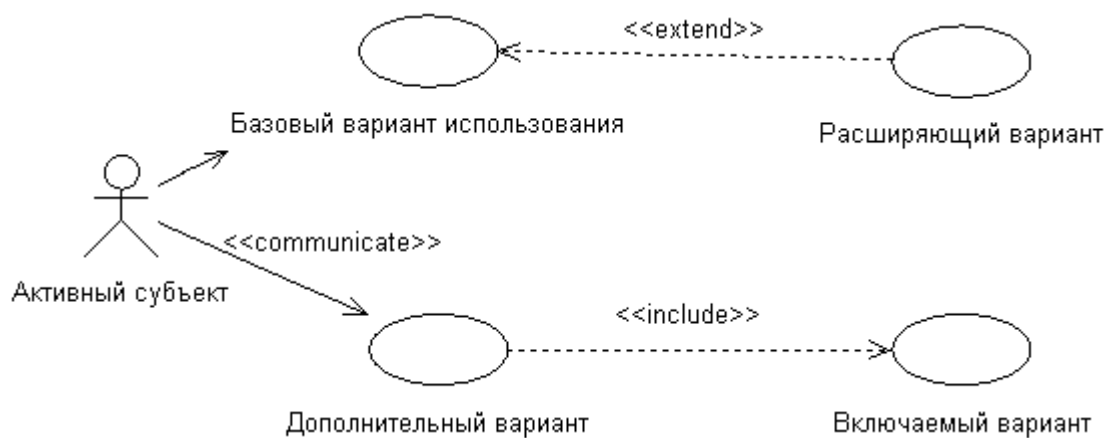


Рисунок 2.1 – Пример диаграммы с ассоциациями и связями зависимости

Разработка диаграммы вариантов использования. *Диаграмма вариантов использования (use case diagram)* – это графическое представление подмножеств активных субъектов, взаимодействующих с системой посредством тех или иных вариантов ее использования. При проектировании любой системы обычно конструируется *основная* диаграмма (**Main Use Case Diagram**), представляющая множество пользователей (активных субъектов) и ключевые функции (варианты использования) системы. При необходимости создаются следующие диаграммы:

- диаграмма, представляющая все варианты использования, инициируемые определенным активным субъектом;
- диаграмма, изображающая варианты использования, подлежащие последовательной реализации;
- диаграмма, описывающая некоторый вариант использования и все его связи.

Таким образом, характеристики поведения разрабатываемой системы фиксируются и документируются средствами модели, которая отображает функции (варианты использования) продукта, представляет окружение системы (множество активных субъектов) и определяет связи между вариантами использования и активными субъектами (диаграммы вариантов использования).

2.2 Методика выполнения работы средствами Rational Rose

При открытии программы **Rational Rose** выводится мастер типовых проектов. Если технология *реализации* проекта еще не выбрана, то от мастера следует отказаться. В результате появится рабочий интерфейс программы с чистым окном активной диаграммы классов и именем проекта **untitled** по умолчанию. Для изменения имени следует сохранить модель во внешнем файле, указав новое имя.

Активизирование диаграммы вариантов использования:

1. Раскрыть представление вариантов использования **Use Case View** в браузере проекта и дважды щелкнуть на пиктограмме **Main** (Главная).
2. С помощью операции главного меню **Browse ► Use Case Diagram** (Браузер ► Диаграмма вариантов использования).

При этом появляется специальная панель инструментов с пиктограммами элементов, назначение которых показано на рисунке 2.2.

Состав инструментов на специальной панели можно изменить с помощью операции главного меню **Tools ► Options** (Инструменты ► Параметры), раскрыв вкладку **Toolbars** (Панель инструментов) и нажав соответствующую кнопку, например, **Use Case Diagram** в группе опций

Customize Toolbars (Настройка панели инструментов). В результате открывается диалоговое окно (рис. 2.3).










	Selection Tool - превращает изображение курсора в форму стрелки
ABC	Text Box - добавляет на диаграмму текстовую область
	Note - добавляет на диаграмму примечание
	Anchor Note to Item - добавляет связь примечания с элементом диаграммы
	Package - добавляет на диаграмму пакет
	Use Case - добавляет на диаграмму вариант использования
	Actor - добавляет на диаграмму актера
	Unidirectional Association - добавляет направленную ассоциацию
	Dependency or Instantiates - добавляет отношение зависимости
	Generalization - добавляет отношение обобщения

Рисунок 2.3 – Назначение кнопок специальной панели инструментов для диаграммы вариантов использования

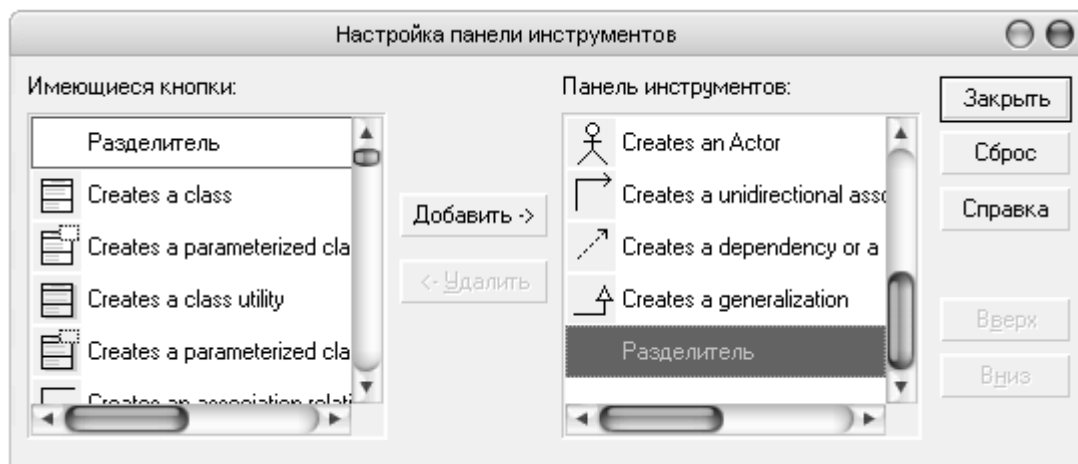


Рисунок 2.2 – Окно настройки специальной панели инструментов для диаграммы вариантов использования

Создание активного субъекта:

1. Расположить курсор мыши над элементом **Use Case View** окна браузера (**Browser**) и щелкнуть правой кнопкой, чтобы активизировать контекстное меню.
2. Выбрать элемент меню **New ► Actor**. Дерево, отображаемое в окне **Browser**, пополнится элементом **NewClass**, соответствующим новому активному субъекту.

3. Изменить название элемента **NewClass**, введя требуемое имя активного субъекта, например. “Клиент”, “Менеджер”, “Оператор”, как показано на рисунке 2.4.

Документирование активного субъекта:

1. Если окно документирования (Documentation) не отображается, открыть его, активизировав элемент меню **View ► Documentation**.
2. В окне **Browser** выбрать элемент дерева, соответствующий требуемому активному субъекту.
3. Разместить курсор в окне **Documentation** и ввести текстовое описание активного субъекта (рис. 2.5).

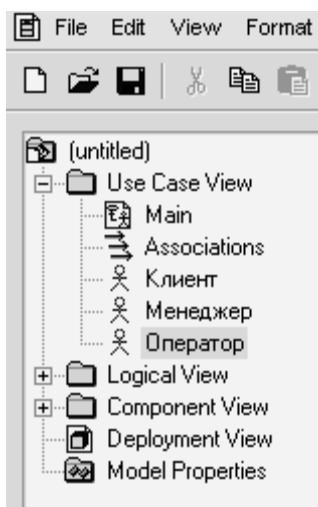


Рисунок 2.4 – Создание активного субъекта

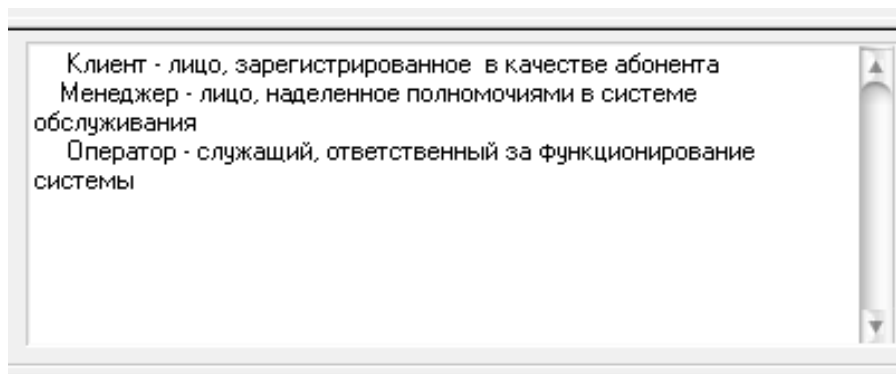


Рисунок 2.5 – Окно документирования активных субъектов

Создание вариантов использования:

1. Расположить курсор мыши над элементом **Use Case View** окна **Browser** и щелкнуть правой кнопкой, чтобы активизировать контекстное меню.
2. Выбрать элемент меню **New ► Use Case**. Дерево, отображаемое в окне **Browser**, пополнится элементом **NewClass**,

соответствующим новому варианту использования.

3. Изменить название элемента **NewClass**, введя требуемое имя варианта использования.

Установление связи документа спецификации потока событий с вариантом использования:

1. Расположить курсор мыши над элементом окна **Browser**, соответствующим требуемому варианту использования, и щелкнуть правой кнопкой, чтобы активизировать контекстное меню.
2. Выбрать элемент меню **Open Specification**.
3. Перейти на вкладку **Files** диалогового окна **Use Case Specification**.
4. Расположить курсор мыши в пределах области окна и щелкнуть правой кнопкой, чтобы активизировать контекстное меню.
5. Выбрать элемент меню **Insert File**.
6. С помощью средств навигации диалогового окна **Открыть** выбрать требуемый файл.
7. Щелкнуть на кнопке **Открыть**.
8. Щелкнуть на кнопке **ОК** окна **Use Case Specification**.
9. Результат связывания документа спецификации с вариантом использования отображается в окне **Browser**.

Создание коммуникативной ассоциации:

1. Щелкнуть на пиктограмме **Unidirectional Association** специальной панели инструментов.
2. В окне диаграммы щелкнуть на символе активного субъекта и, не отпуская кнопку мыши, построить линию связи, направленную к символу соответствующего варианта использования.

Создание метки стереотипа «communicate»:

1. В окне диаграммы дважды щелкнуть на линии, представляющей ассоциацию, чтобы открыть диалоговое окно **Association Specification**.
2. Щелкнуть на кнопке со стрелкой в правой части поля **Stereotype** и выбрать в раскрывающемся списке опцию **communicate**.
3. Закрыть окно **Association Specification** щелчком на кнопке **ОК**.

Создание включающей связи:

1. Щелкнуть на пиктограмме **Dependency or Instantiates** специальной панели инструментов.
2. В окне диаграммы щелкнуть на символе базового варианта использования и, не отпуская кнопку мыши, построить линию связи, направленную к символу соответствующего включаемого варианта использования.
3. Дважды щелкнуть на линии, представляющей связь, чтобы

- открыть диалоговое окно **Dependency Specification**.
- Щелкнуть на кнопке со стрелкой в правой части поля **Stereotype** и выбрать в раскрывающемся списке опцию **include**.
 - Закрыть окно **Dependency Specification** щелчком на кнопке **ОК**.

Создание расширяющей связи:

- Щелкнуть на пиктограмме **Dependency or Instantiates** специальной панели инструментов.
- В окне диаграммы щелкнуть на символе расширяющего варианта использования и, не отпуская кнопку мыши, построить линию связи, направленную к символу соответствующего базового варианта использования.
- Дважды щелкнуть на линии, представляющей связь, чтобы открыть диалоговое окно **Dependency Specification**.
- Щелкнуть на кнопке со стрелкой в правой части поля **Stereotype** и выбрать в раскрывающемся списке опцию **extend**.
- Закрыть окно **Dependency Specification** щелчком на кнопке **ОК**.

Создание основной диаграммы вариантов использования:

- Двойным щелчком на элементе **Main** поддерева **Use Case View**, отображаемого в окне **Browser**, открыть окно основной диаграммы вариантов использования.
- В окне **Browser** выбрать элемент, соответствующий требуемому активному субъекту, и перетащить его в окно диаграммы.
- Повторить действие, указанное в п. 2, для всех активных субъектов, подлежащих включению в диаграмму.
- В окне **Browser** выбрать элемент, соответствующий требуемому варианту использования, и перетащить его в окно диаграммы.
- Повторить действие, указанное в п. 4, для всех вариантов использования, подлежащих включению в диаграмму.

В качестве примера на рисунке 2.6 изображена основная диаграмма вариантов использования для системы учета успеваемости студентов.

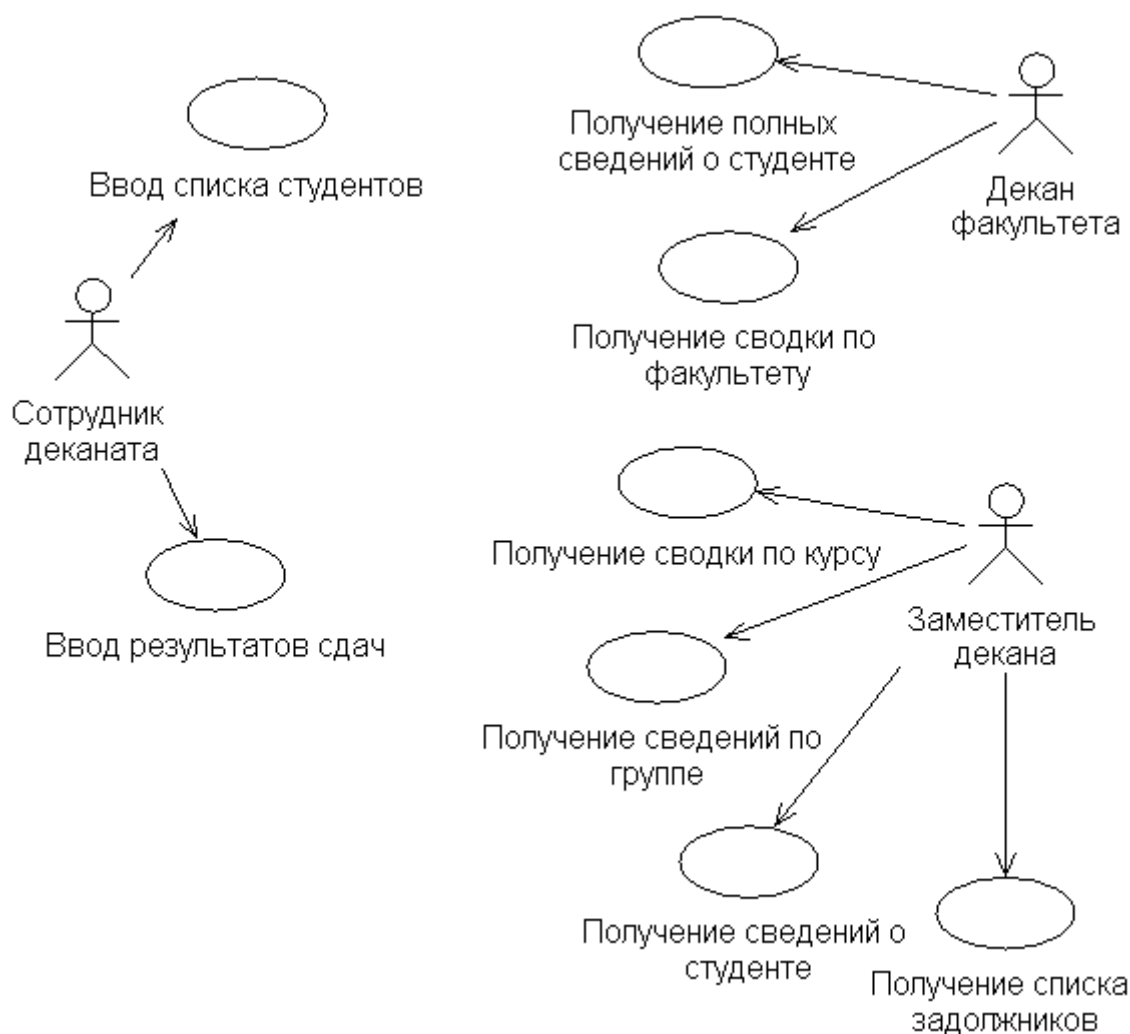


Рисунок 2.6 – Диаграмма вариантов использования системы учета успеваемости студентов

Содержание отчета

Отчет по работе должен содержать основную диаграмму вариантов использования с документированием активных субъектов, потоков событий и связей. При защите отчета необходимо обосновать варианты использования, а также выбор актеров, типов связей, основного и альтернативных потоков событий.

3 КОНСТРУИРОВАНИЕ КЛАССОВ

Цель лабораторной работы: освоение приемов и методики конструирования классов с применением программы Rational Rose.

3.1 Рекомендации по определению и конструированию классов

Класс (class) – это группа объектов с общими свойствами (атрибутами), поведением (функциями), семантикой и связями с другими объектами. Класс можно рассматривать как шаблон для создания объектов. Каждый объект является экземпляром *только одного класса*.

Объект (object) – это представление реальной или абстрактной сущности – автомобиля, персонального компьютера, металлорежущего станка, электродвигателя, датчика или банковской транзакции. Объект представляет собой понятие с явно оговоренными границами, смыслом и назначением в контексте программного приложения. Каждый объект системы обладает тремя характеристиками – *состоянием, поведением и идентификационным признаком*.

Состояние (state) объекта – это одно из возможных сочетаний условий его существования. Состояние объекта определяется набором *свойств-атрибутов (attributes)* и связей с другими объектами.

Поведение (behavior) охватывает функциональную сторону жизни объекта, определяет его реакцию на запросы со стороны других объектов и реализуется в виде набора операций.

Идентификационный признак (identity) задает свойство уникальности объекта – даже в том случае, если состояние последнего идентично состоянию других объектов.

Правильно сконструированный класс должен представлять одну и *только одну абстракцию*. Для именованых классов следует пользоваться терминами, принятыми в соответствующей предметной области. В качестве имени класса принято употреблять существительное единственного числа, наилучшим образом описывающее моделируемое понятие.

Классам отвечают определенные стереотипы, позволяющие создавать новые разновидности элементов моделируемой системы. К числу наиболее употребительных стереотипов классов относятся:

- *entity* – сущность;
- *boundary* – граница;
- *control* – управление;
- *utility* – прикладной класс;
- *exception* – исключение.

Названия стереотипов на диаграммах заключаются в угловые кавычки (<<...>>) и располагаются над именами классов. Система Rational Rose позволяет ассоциировать со стереотипом определенную пиктограмму или

выделять его тем или иным цветом. Стереотипы позволяет аналитику и дизайнеру размежевать уровни предметной области.

Класс сущностей (entity class) моделирует структуру данных со стабильным характером. Класс подобного типа отражает качества сущности реального мира или применяется для выполнения внутренних функций системы. Классы сущностей обычно не зависят от окружения, они не чувствительны к внешним обстоятельствам. Для описания определенной функции подойдет краткое выражение с именем существительным. Классы сущностей часто называют "предметными", поскольку они представляют абстракции понятий и вещей реального мира.

Классы границ (boundary classes) обслуживают процессы взаимодействия между системой и ее окружением, обеспечивая интерфейсы для пользователей и сторонних систем (активных субъектов). Такие классы образуют ту часть системы, которая непосредственно "общается" с внешним миром. Для отыскания классов границ анализируется каждая пара вида "активный субъект/вариант использования". Классы границ определяются с учетом особенностей выбранных механизмов интерфейса.

Классы управления (control classes) применяются при реализации поведения системы и координируют события, возникающие по мере функционирования системы в рамках вариантов использования. Класс управления можно воспринимать как абстракцию, отображающую динамику варианта использования системы. Классы управления обычно тесно "привязаны" к особенностям конкретного приложения.

На ранних стадиях жизненного цикла системы для каждой пары вида "активный субъект/вариант использования" создается по одному классу управления, на который возлагаются обязанности по контролю за потоком событий, происходящих по мере выполнения этого варианта.

Класс управления должен быть "осведомлен" о том, *когда* следует выполнить операцию, но то, *как* это делать, является исключительной прерогативой класса сущностей.

При создании классов их следует документировать. Описание должно характеризовать назначение класса, а не его структуру. При именовании или документировании класса возникают следующие ситуации:

- Описание класса совпадает с описанием другого класса – следует изучить возможность объединения классов.
- Чрезмерно велик комментарий – следует применить расщепление класса.
- Нельзя подобрать ни имя класса, ни вразумительный комментарий к нему – необходим дополнительный анализ, который позволит обосновать или пересмотреть абстракцию.

Если система содержит большое количество классов, они группируются в пакеты. **Пакет** – это собрание "родственных" вложенных пакетов и/или классов. Распределение классов по пакетам дает возможность подняться на более высокий уровень восприятия модели.

Каждый пакет обычно содержит интерфейс, образованный множеством *общедоступных* классов. К таким классам разрешено обращаться из кода классов, принадлежащих другим пакетам. Остальные классы пакета связаны со спецификой реализации и потому не допускают обращения извне.

Если проектируемая система отличается особой сложностью, пакеты могут конструироваться уже на ранних фазах ее жизненного цикла. В простых ситуациях созданные классы группируются в единственный пакет, "границы" которого определяются системой в целом.

Подмножество пакетов и классов представляются в удобном графическом виде – **диаграммы классов (class diagrams)**.

Основная диаграмма классов (Main Class Diagram) модели изображает, как правило, *набор пакетов системы*. При необходимости создаются и дополнительные диаграммы классов, в том числе и такие, которые связаны с определенными вариантами использования, например:

- диаграмма, представляющая все классы, связанные с внутренней реализацией системы;
- диаграмма, описывающая структуру и поведение одного или нескольких классов;
- диаграмма, изображающая иерархию наследования классов.

3.2 Методика конструирования и документирования классов в среде Rational Rose

Активизирование диаграммы классов:

При создании нового проекта окно диаграммы классов появляется по умолчанию в рабочем окне. В других случаях активизировать рабочее окно диаграммы классов можно несколькими способами:

1. Щелкнуть на кнопке с изображением диаграммы классов на стандартной панели инструментов.
2. Раскрыть логическое представление **Logical View** в браузере проекта и дважды щелкнуть на пиктограмме **Main** (Главная).
3. С помощью операции главного меню **Browse ► Class Diagram** (Браузер ► Диаграмма классов).

При этом появляется новое окно с чистым листом и специальная панель инструментов с пиктограммами элементов, назначение которых показано на рисунке 3.1.

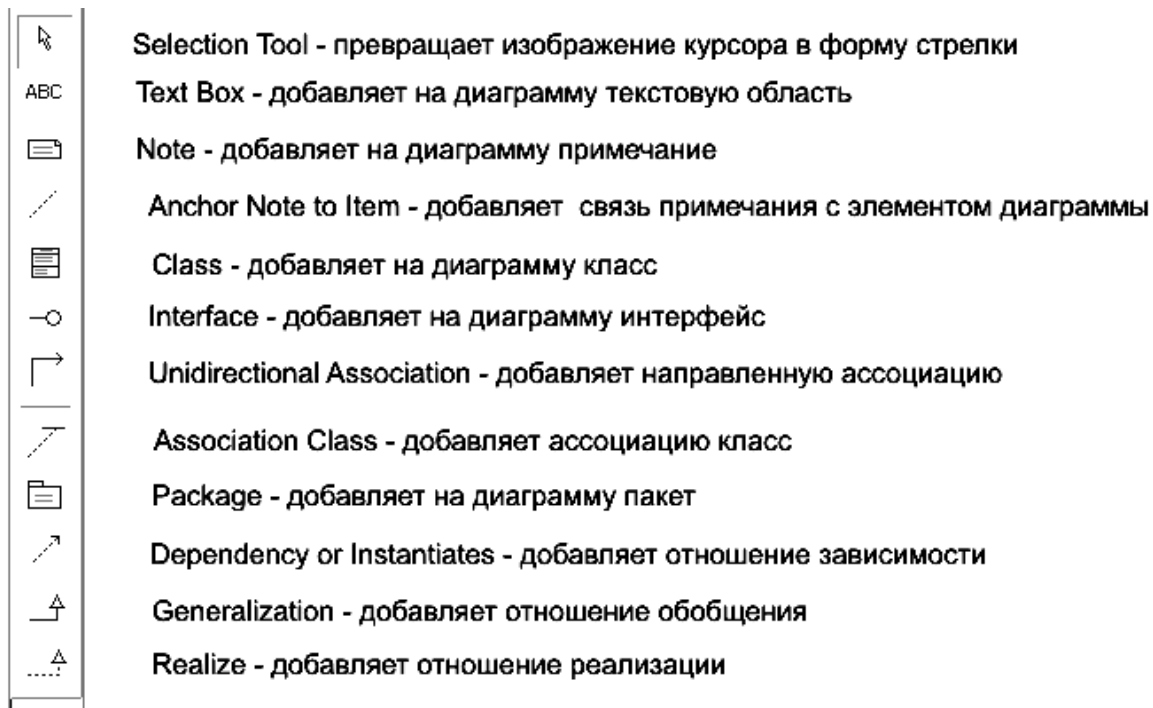


Рисунок 3.1 – Назначение кнопок специальной панели инструментов для диаграммы классов

Добавление класса на диаграмму:

1. Расположить курсор мыши над элементом **Logical View** окна **Browser** и щелкнуть правой кнопкой, чтобы активизировать контекстное меню.
2. Выбрать элемент меню **New ► Class**. При этом дерево, отображаемое в окне **Browser**, пополнится элементом << >> **NewClass**.
3. Изменить название элемента **NewClass**, введя требуемое имя класса.

Класс, добавленный на рабочую страницу, отображается в окне **Browser** с соответствующей пиктограммой стереотипа, введенными атрибутами и операциями. Пример отображения класса **Счет** показан на рисунке 3.2.

Создание стереотипа класса:

1. Расположить курсор мыши над элементом окна **Browser**, соответствующим требуемому классу, и щелкнуть правой кнопкой, чтобы активизировать контекстное меню.
2. Выбрать элемент меню **Open Specification**.
3. Перейти на вкладку **General** диалогового окна **Class Specification**.
4. Щелкнуть на кнопке со стрелкой в правой части поля **Stereotype**, чтобы раскрыть список опций, и выбрать нужную опцию либо создать новый стереотип, введя в поле соответствующее наименование.

5. Закрывать окно **Class Specification** щелчком на кнопке **OK**.

Диалоговое окно **Class Specification**, отвечающее классу **Счет**, показано на рисунке 3.3.

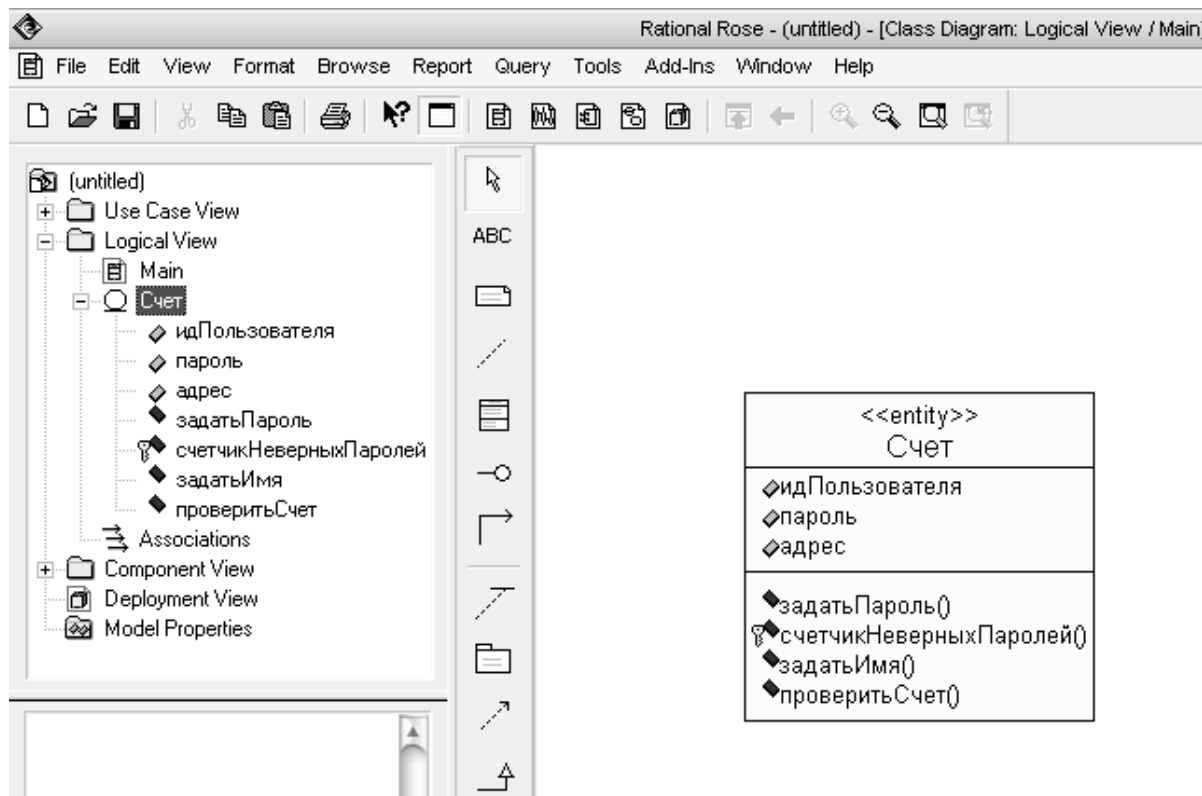


Рисунок 3.2 – Рабочее окно и браузер после добавления диаграммы класса **Счет**

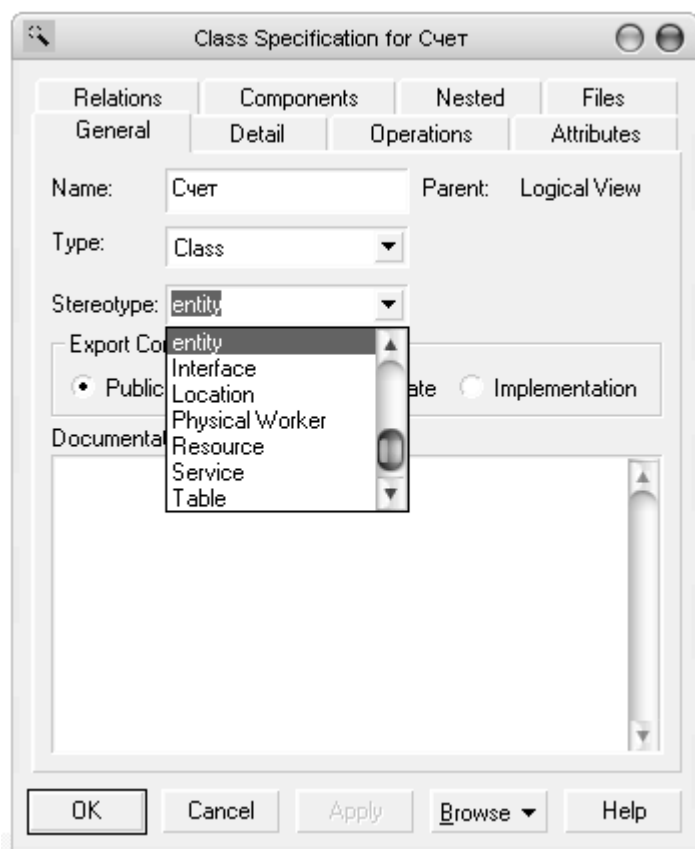
Документирование класса:

1. Если окно документирования (**Documentation**) не отображается, открыть его, активизировав элемент меню **View ► Documentation**.
2. В окне **Browser** выбрать элемент дерева, соответствующий требуемому классу.
3. Позиционировать курсор в окне **Documentation** и ввести текстовое описание класса.

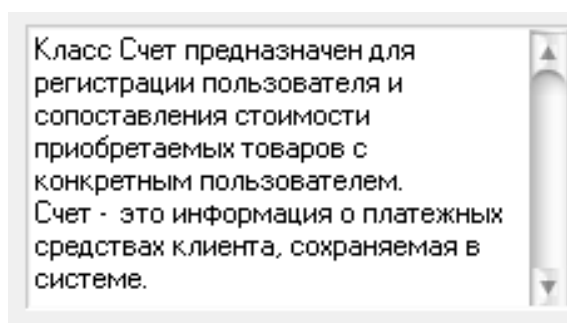
Окно **Documentation** с текстом, документирующим класс **Счет**, показано на рисунке 3.4.

Создание пакета:

1. Расположить курсор мыши над элементом **Logical View** окна **Browser** и щелкнуть правой кнопкой, чтобы активизировать контекстное меню.
2. Выбрать элемент меню **New ► Package**. При этом дерево, отображаемое в окне **Browser**, пополнится элементом **NewPackage**, соответствующим новому пакету.
3. Изменить название **NewPackage**, введя требуемое имя пакета.



*Рисунок 3.3 – Диалоговое окно спецификации свойств класса **Счет** при выборе из вложенного списка стереотипа **entity***



*Рисунок 3.4 – Окно документирования класса **Счет***

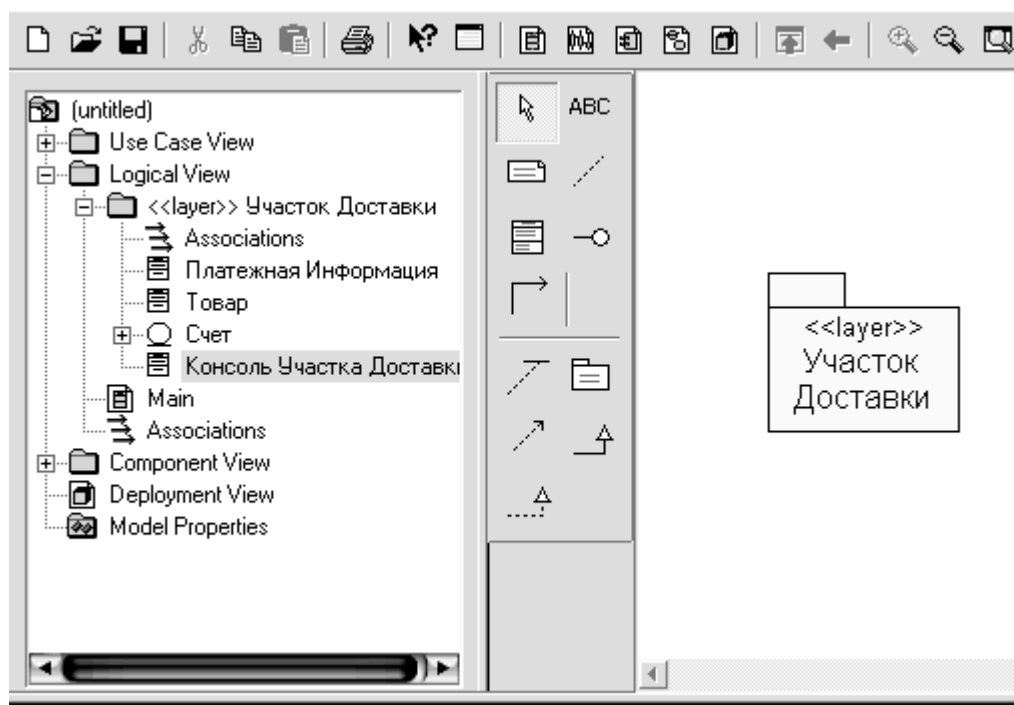
Размещение класса в пакете:

В окне **Browser** выбрать элемент дерева, соответствующий требуемому классу.

Перетащить и опустить его на элемент, представляющий нужный пакет.

Повторить операцию для каждого класса, подлежащего перемещению.

Окно браузера после распределения классов по пакетам показано на рисунке 3.5.



*Рисунок 3.5 – Создание пакета **Участок Доставки** из классов **Платежная Информация, Товар, Счет, Консоль Участка Доставки** и его отображение в браузере*

Установка признака принадлежности класса пакету:

1. Чтобы задать значение признака, предлагаемое по умолчанию, следует выбрать элемент меню **Tools ► Options**.
2. Перейти на вкладку **Diagram** диалогового окна **Options**.
3. В группе опций **Compartments** установить флажок **Show visibility** и закрыть окно щелчком на кнопке **ОК** (рис. 3.6).
4. Чтобы задать значение признака для конкретного класса, расположить курсор мыши над элементом, представляющим соответствующий класс в окне диаграммы классов, и щелкнуть правой кнопкой, активизируя контекстное меню.
5. Установить или сбросить признак с помощью элемента меню **Options Show Visibility**.

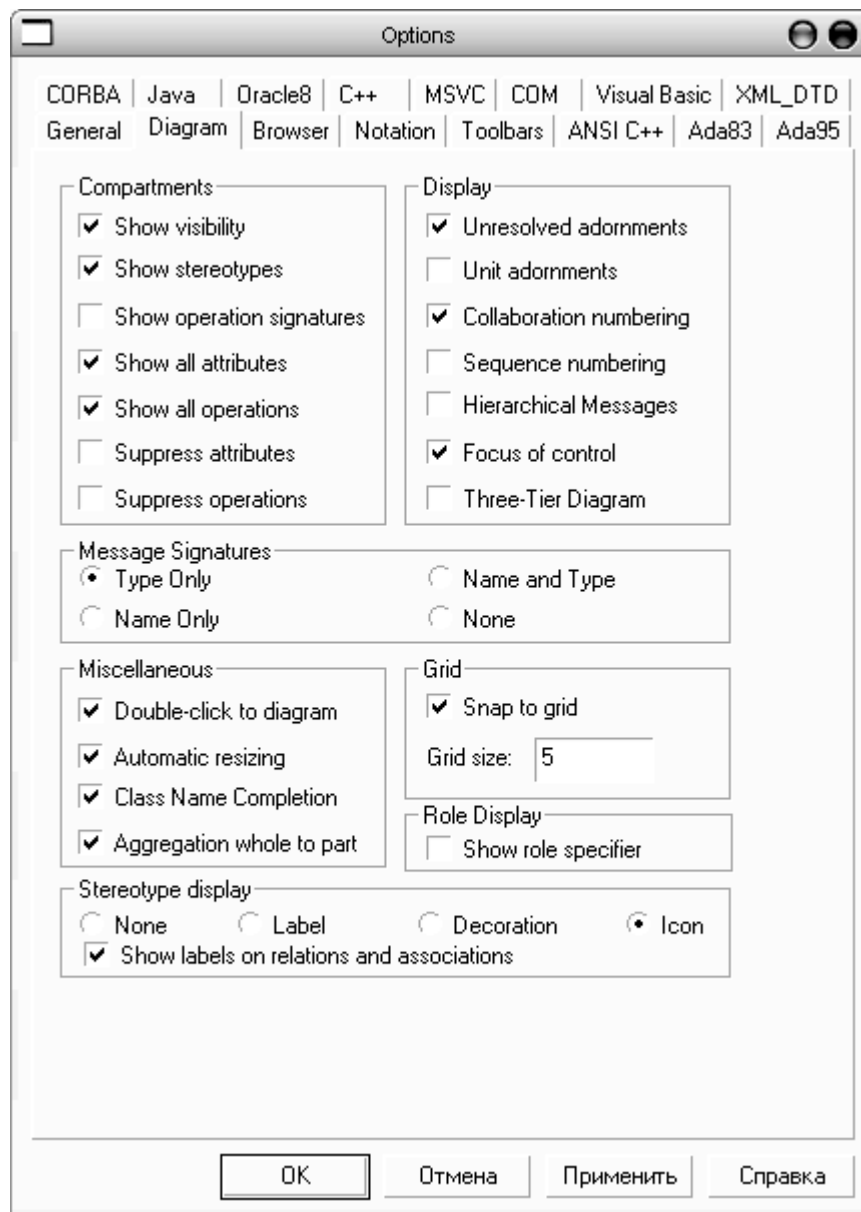


Рисунок 3.6 – Установка признака принадлежности класса пакету

Изображение пакета приобретает вид, показанный на рисунке 3.7.

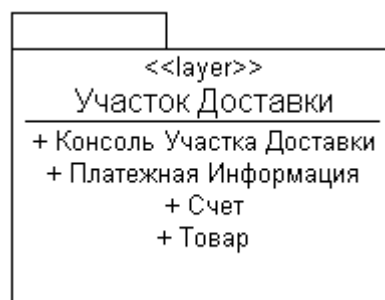


Рисунок 3.7 – Пакет с отображением включенных в него классов

Создание основной диаграммы классов (диаграммы пакетов):

Открыть окно основной диаграммы классов двойным щелчком на элементе **Main** поддерева **Logical View**, отображаемого в окне **Browser**.

В окне **Browser** выбрать элемент, соответствующий требуемому пакету, и перетащить его в окно диаграммы.

Повторить действие, указанное в п. 2, для всех пакетов, подлежащих включению в диаграмму.

Операции по размещению пакетов на диаграмме можно выполнить также средствами специальной панели инструментов.

Диаграмма классов, относящихся к нескольким пакетам, приведена на рисунке 3.8.

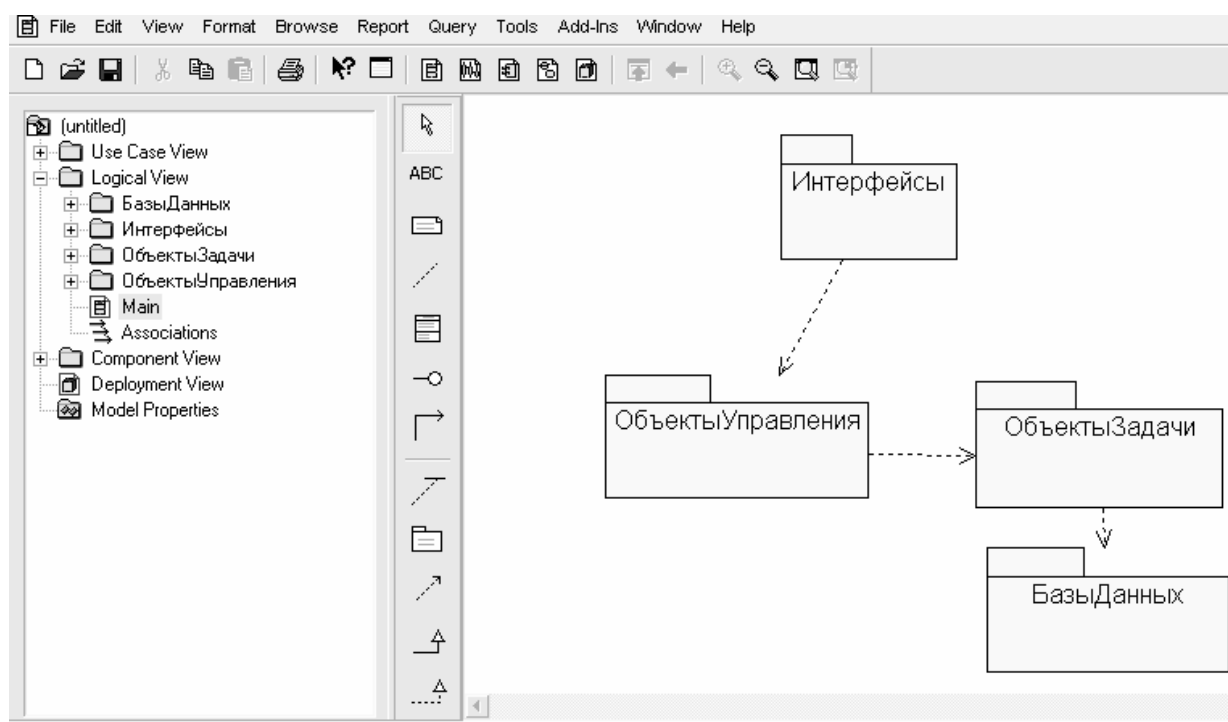


Рисунок 3.8 – Рабочее окно с диаграммой пакетов и браузер при создании основной диаграммы классов для системы решения комбинаторных задач

3.3 Содержание отчета

Отчет по работе должен содержать диаграмму классов с документированием назначения классов, их атрибутов, операций и стереотипов, а также диаграмму пакетов. При защите отчета необходимо объяснить приемы работы с программой, а также обосновать принятые в модели решения.

4 РАЗРАБОТКА ДИАГРАММ ВЗАИМОДЕЙСТВИЯ

Цель лабораторной работы: освоение приемов и методики разработки диаграмм взаимодействия с применением программы Rational Rose.

4.1 Рекомендации по созданию диаграмм взаимодействия

Функции, охватываемые вариантом использования, фиксируются в потоке событий, а для описания способов реализации вариантов использования применяются сценарии.

Сценарий (scenario) – это экземпляр варианта использования, один из возможных потоков событий для этого варианта. Сценарии помогают идентифицировать объекты, разработать адекватные классы и выявить примеры взаимодействия объектов *в процессе выполнения функций*, предусмотренных вариантом использования. Сценарии документируют решения о том, каким образом функции, возлагаемые на вариант использования, распределяются между объектами и классами системы.

Поток событий для варианта использования описывается в текстовом виде, а сценарии представляются в форме *диаграмм взаимодействия (interaction diagrams)*. Различают два типа таких диаграмм – *диаграммы последовательностей (sequence diagrams)* и *диаграммы сотрудничества (collaboration diagrams)*.

Диаграмма последовательностей иллюстрирует очередность выполнения операций взаимодействия объектов во времени и отображает объекты и классы, вовлеченные в сценарий. На диаграммах последовательностей показываются также сообщения, которыми объекты обмениваются в ходе осуществления функций, предусмотренных сценарием. Диаграммы последовательностей обычно ассоциируются с реализациями вариантов использования, перечисленными в пакете **Logical View**.

В соответствии с требованиями UML объект на диаграмме последовательностей изображается в виде прямоугольника, содержащего *подчеркнутое наименование объекта*. Объект можно именовать тремя способами: указать только его название, задать имена объекта и класса либо ограничиться наименованием класса (для анонимного объекта).

Каждому объекту на диаграмме последовательностей ставится в соответствие временная отметка, обозначаемая отрезком штриховой линии, а сообщения, которыми обмениваются два объекта, представляются стрелкой, соединяющей объект-источник с объектом-приемником.

В диаграммы последовательностей целесообразно включать граничные классы, позволяющие отобразить факты взаимодействия системы с активными субъектами (пользователями и сторонними системами). На ранних стадиях анализа такой прием позволяет зафиксировать и документировать требования к интерфейсам.

При разработке диаграмм последовательностей следует стремиться к простоте их представления, что позволяет сразу видеть и объекты, и

сообщения, которыми объекты обмениваются, и функции, охватываемые сценарием.

Если на диаграммах последовательностей должны быть реализованы условные логические выражения *если-то* и *иначе*, которые воспроизводят реалии предметной области, то их можно включить в одну диаграмму, используя дополнительные примечания для описания возможных вариантов выбора. Однако лучше создавать отдельные диаграммы – одну для случая *если-то*, а другую для иллюстрации ветви *иначе*. Это позволит упростить восприятие модели.

4.2 Методика разработки диаграмм взаимодействия

Активизировать рабочее окно диаграммы последовательностей можно следующими способами:

- Щелкнуть на кнопке с изображением диаграммы взаимодействия на стандартной панели инструментов (**Browse Interaction Diagram**), в открывшемся окне **Select Interaction Diagram** дважды щелкнуть левой кнопкой мыши на поле **<New>** и в открывшемся диалоговом окне **New Interaction Diagram** (рис. 4.1) выбрать диаграмму последовательностей (**Sequence**) и внести имя диаграммы в поле **Title**.

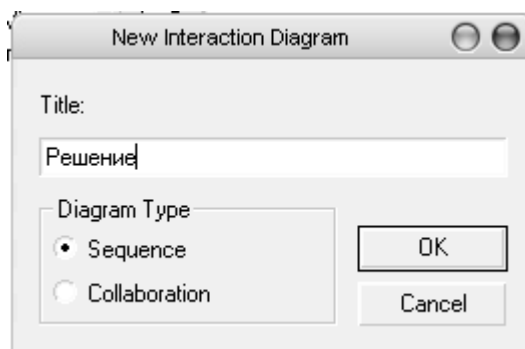


Рисунок 4.1 – Выбор диаграммы последовательностей и внесение её наименования

- Расположить курсор мыши над элементом **Logical View** окна **Browser**, активизировать правой кнопкой контекстное меню, выбрать элементы меню **New ► Sequence Diagram**, как показано на рисунке 4.2. Далее внести имя диаграммы в окне **Browser** и дважды щелкнуть по нему левой кнопкой мыши.

После выбора диаграммы последовательностей вместо специальной панели инструментов для диаграммы классов появляется специальная панель инструментов, содержащая кнопки с изображениями примитивов, необходимые для разработки диаграммы последовательностей (рис. 4.3).

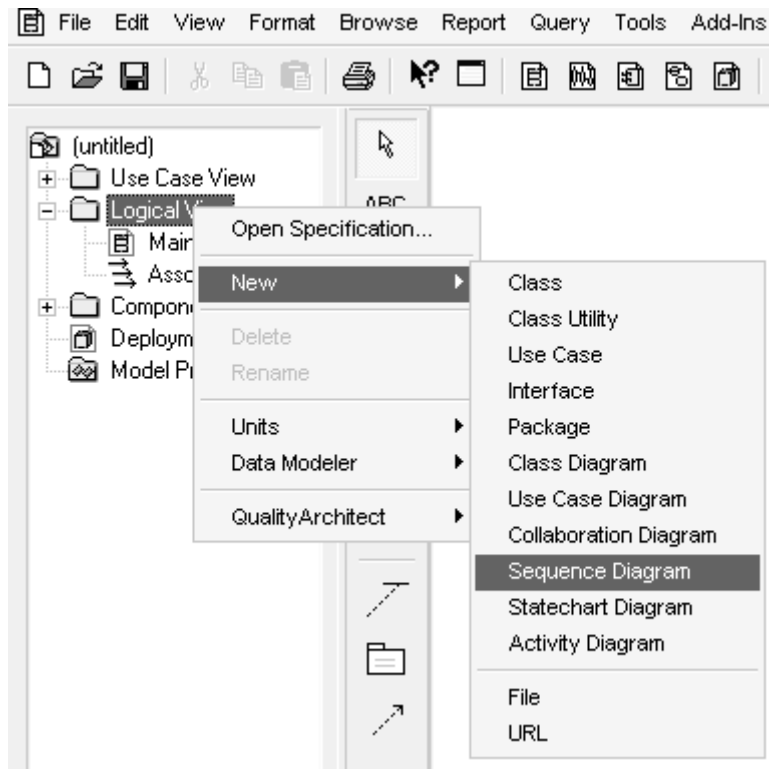


Рисунок 4.2 – Выбор диаграммы последовательностей с помощью контекстного меню

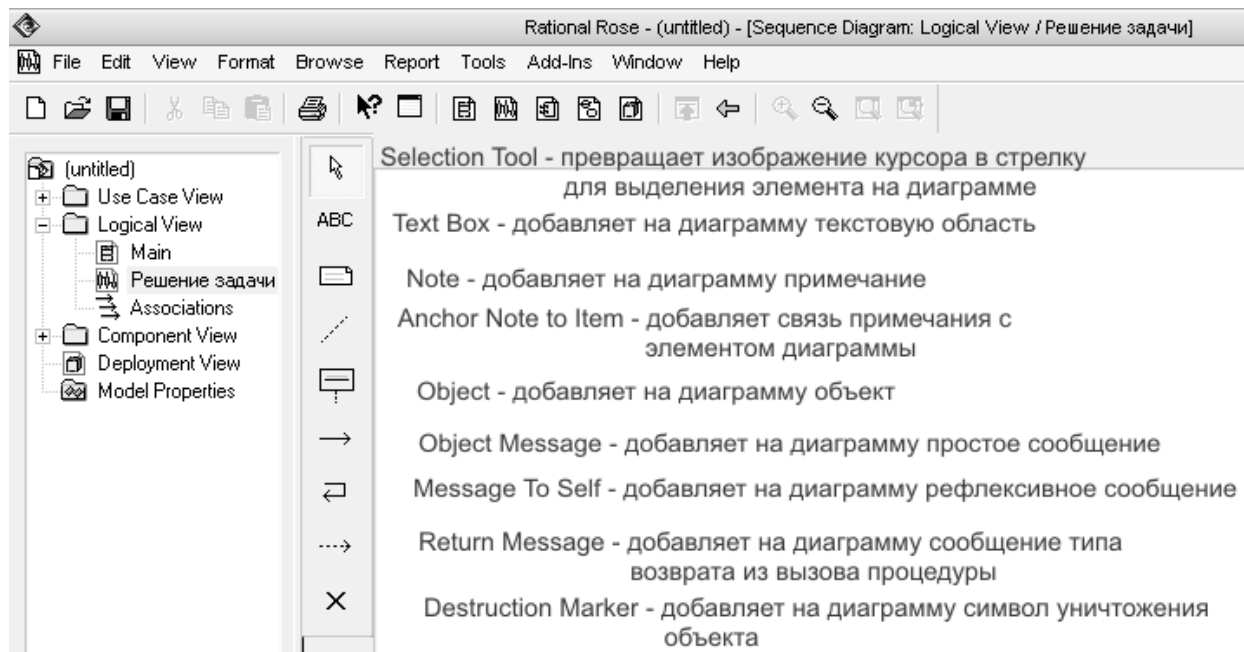


Рисунок 4.3 – Специальная панель инструментов для разработки диаграммы последовательностей

Создание объектов и сообщений в диаграмме последовательностей:

1. В окне **Browser** выбрать элемент, соответствующий требуемому

- активному субъекту, и перетащить его в рабочее окно диаграммы.
2. Щелкнуть на пиктограмме **Object** специальной панели инструментов.
 3. Щелкнуть в соответствующей позиции рабочей области окна диаграммы, чтобы разместить в ней новый объект.
 4. Выбрать объект на диаграмме и ввести его имя в окне спецификации объекта.
 5. Повторить действия, указанные в пп. 2-4, для всех объектов, участвующих в реализации сценария.
 6. Щелкнуть на пиктограмме **Object Message** специальной панели инструментов.
 7. В окне диаграммы щелкнуть на символе активного субъекта (объекта), служащего источником, и, не отпуская кнопку мыши, построить линию сообщения, направленную к символу соответствующего активного субъекта (объекта), который является приемником.
 8. Дважды щелкнуть на линии, представляющей сообщение, чтобы открыть диалоговое окно **Message Specification**, в поле **Name** ввести текст сообщения и закрыть окно щелчком на кнопке **OK**.
 9. Повторить действия, указанные в пп. 6-8, для всех сообщений, инициируемых при реализации сценария.

Диаграмма последовательностей для сценария "Решение задачи" изображена на рисунке 4.4.

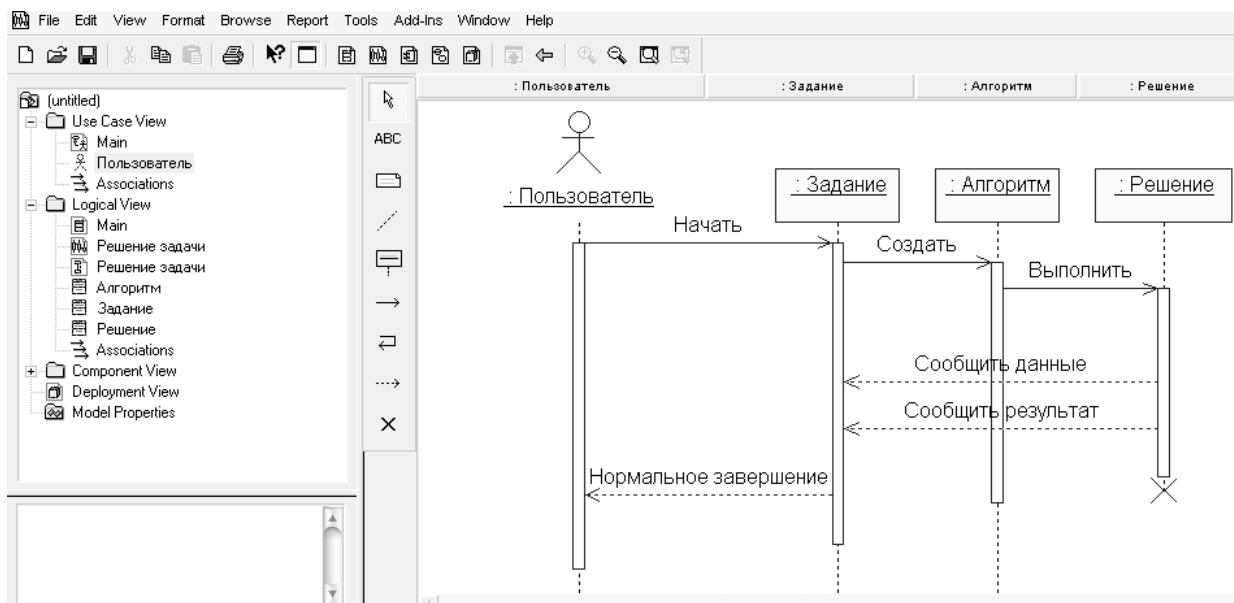


Рисунок 4.4 – Диаграмма последовательностей для сценария "Решение задачи"

Каждый добавляемый на диаграмму объект по умолчанию считается анонимным – выводится двоеточие и наименование класса. Для внесения

имени объекта следует вызвать диалоговое окно свойств объекта (рис. 4.5), дважды щелкнув на его изображении.

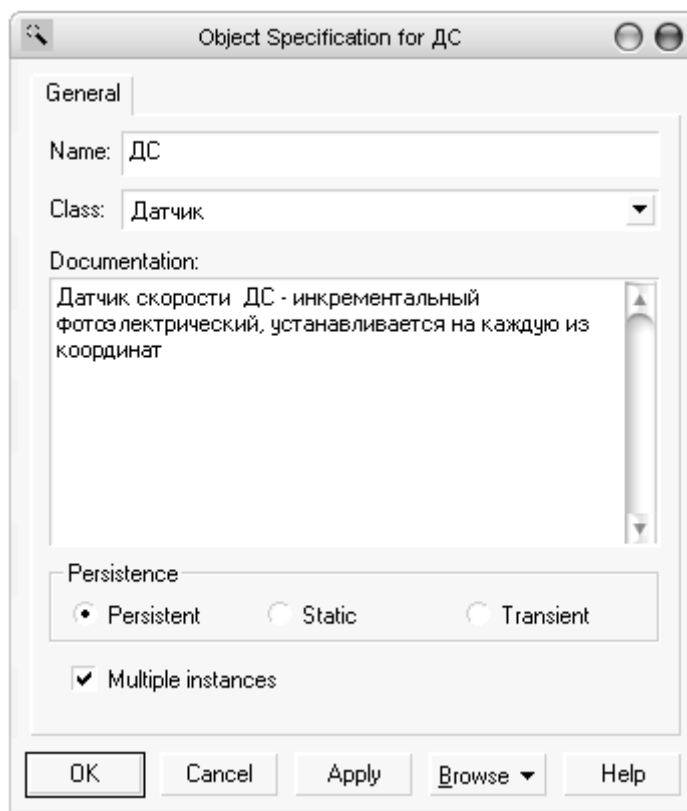


Рисунок 4.5 – Диалоговое окно спецификации свойств объекта класса Датчик

В окне спецификации свойств после введения имени, например, датчика скорости (ДС), необходимо также задать свойства устойчивости (**Persistence**) и множественности (**Multiple instances**).

Свойство **Persistent** (Устойчивый) означает, что информация об объектах данного класса должна сохраняться постоянно, независимо от используемого приложения.

Свойство **Static** (Статический) означает, что соответствующий объект хранится в памяти в течение работы программного приложения.

Свойство **Transient** (Временный) означает, что соответствующий объект хранится в памяти системы в течение короткого времени, необходимого только для выполнения операции.

Если в системе используется несколько экземпляров, следует установить метку **Multiple instances**.

Создание диаграммы сотрудничества на основе диаграммы последовательностей:

1. При открытом окне диаграммы последовательностей выбрать элемент меню **Browse ► Create Collaboration Diagram** или нажать клавишу **<F5>**.

2. При необходимости упорядочить объекты и сообщения диаграммы сотрудничества.

Диаграмма сотрудничества для сценария “Решение задачи” показана на рисунке 4.6.

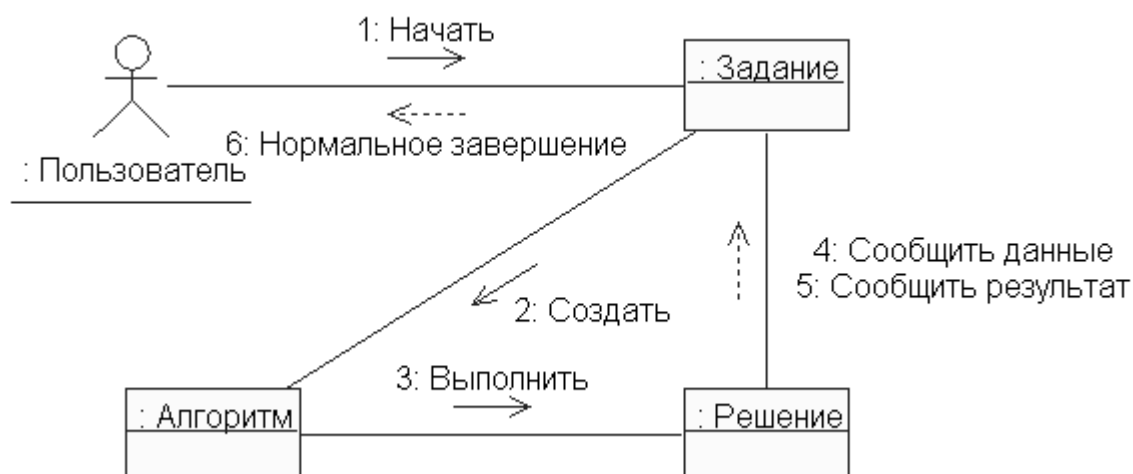


Рисунок 4.6 – Диаграмма сотрудничества, полученная из диаграммы последовательности (рис. 4.4)

4.3 Содержание отчета

Отчет по работе должен содержать диаграммы последовательности и сотрудничества, снабженные соответствующими связями и спецификациями.

При защите отчета необходимо объяснить приемы работы с программой Rational Rose, а также обосновать принятые решения.