

4. Разработка диаграммы классов и редактирование их свойств

Особенности разработки диаграмм классов в среде IBM Rational Rose 2003. Добавление класса на диаграмму классов и редактирование его свойств. Стереотипы класса сущности и управляющего класса и их графическое представление. Графические стереотипы граничного класса и интерфейса.










Особенности разработки диаграмм классов в среде IBM Rational Rose 2003




Диаграмма *классов* является основным логическим представлением модели и содержит детальную информацию о внутреннем устройстве объектно-ориентированной программной системы или, используя современную терминологию, об архитектуре программной системы. Активизировать рабочее окно диаграммы *классов* можно несколькими способами:

- окно диаграммы *классов* появляется по умолчанию в рабочем окне диаграммы после создания нового проекта;
- щелкнуть на кнопке с изображением диаграммы *классов* на стандартной панели инструментов;
- раскрыть логическое представление (Logical View) в браузере проекта и дважды щелкнуть на пиктограмме **Main** (Главная);
- выполнить операцию главного меню: **Browse** → **Class Diagram** (Обзор → Диаграмма классов).

При этом появляется новое окно с чистым рабочим листом диаграммы *классов* и специальная панель инструментов, содержащая кнопки с изображением графических примитивов, необходимых для разработки диаграммы *классов* (табл. 4.1). Назначение отдельных кнопок панели можно узнать также из всплывающих подсказок.

Таблица 4.1. Назначение кнопок специальной панели инструментов для диаграммы классов

Графическое изображение	Всплывающая подсказка	Назначение кнопки
	Selection Tool	Превращает изображение курсора в форму стрелки для последующего выделения элементов на диаграмме
	Text Box	Добавляет на диаграмму текстовую область
	Note	Добавляет на диаграмму примечание
	Anchor Note to Item	Добавляет на диаграмму связь примечания с соответствующим графическим элементом диаграммы
	Class	Добавляет на диаграмму <i>класс</i>
	Interface	Добавляет на диаграмму <i>интерфейс</i>
	Unidirectional Association	Добавляет на диаграмму направленную <i>ассоциацию</i>
	Association Class	Добавляет на диаграмму <i>ассоциацию класс</i>
	Package	Добавляет на диаграмму пакет

		Dependency or Instantiates	Добавляет на диаграмму отношение зависимости
		Generalization	Добавляет на диаграмму отношение обобщения
		Realize	Добавляет на диаграмму отношение реализации

На специальной панели инструментов по умолчанию присутствует только часть пиктограмм элементов, которые могут быть использованы для построения диаграммы *классов*. Добавить кнопки с пиктограммами других графических элементов таких как, например, отношения агрегации и композиции, шаблон, *класс* бизнес-сущность, *управляющий класс*, или удалить ненужные кнопки можно с помощью настройки специальной панели инструментов. Соответствующее диалоговое окно настройки специальной панели инструментов для диаграммы *классов* можно вызвать аналогично другим панелям с помощью операции контекстного меню **Customize** (Настройка) при позиционировании курсора на специальной панели инструментов.

Добавление класса на диаграмму классов и редактирование его свойств

Для добавления *класса* на диаграмму *классов* нужно с помощью левой кнопки мыши нажать кнопку с изображением пиктограммы *класса* на специальной панели инструментов, отпустить левую кнопку мыши и щелкнуть левой кнопкой мыши на свободном месте рабочего листа диаграммы. На диаграмме появится изображение *класса* с маркерами изменения его геометрических размеров и предложенным средой именем по умолчанию **NewClass**.

Продолжая разработку модели банкомата в качестве сквозного примера проекта, построим для этой модели следующую каноническую диаграмму - диаграмму *классов*. С этой целью следует изменить предложенное по умолчанию имя диаграммы **Main** на **Диаграмма классов АТМ**, а имя добавленного на диаграмму *класса* - на **Транзакция Банкомата** (рис. 4.1).

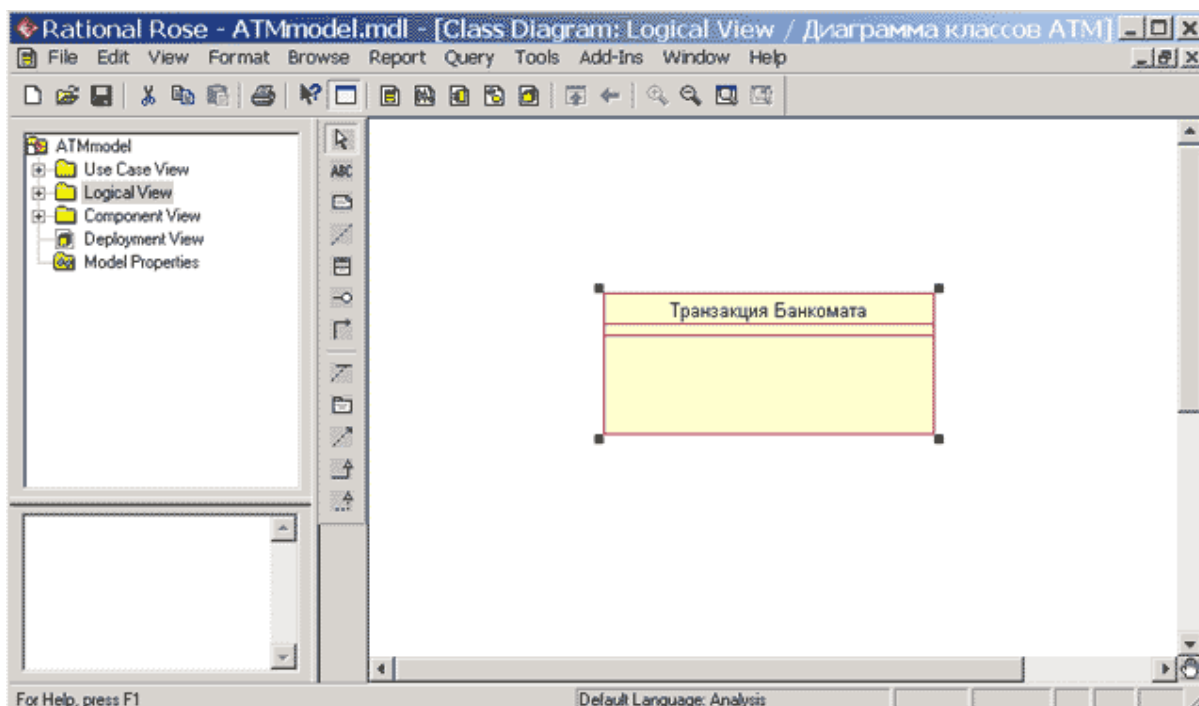


Рис. 4.1. Диаграмма классов модели банкомата после добавления на нее класса Транзакция Банкомата

Поскольку разрабатываемая модель банкомата на начальных этапах работы над проектом используется для анализа общей архитектуры проекта и согласования ее с различными участниками рабочей группы, имена *классов*, их атрибутов и операций для большей наглядности и понимания задаются на русском языке с пробелами и записываются символами кириллицы. В последующем по мере выполнения проекта и реализации модели на некотором языке программирования, имена соответствующих *классов*, атрибутов и операций должны быть преобразованы в символы латиницы. При этом имена этих элементов модели должны быть записаны без пробелов. В контексте управляемой моделью архитектуры первую модель еще называют независимой от платформы реализации, а вторую - зависимой от платформы реализации.

Для *класса* **Транзакция Банкомата** можно уточнить его назначение в модели с помощью указания стереотипа и пояснительного текста в форме документации. С этой целью двойным щелчком левой кнопкой мыши на изображении этого *класса* на диаграмме или в браузере проекта следует открыть диалоговое окно спецификации свойств этого *класса* (рис. 4.2) и на вкладке **General** (Общие) выбрать из вложенного списка **Stereotype** стереотип **entity** (сущность).

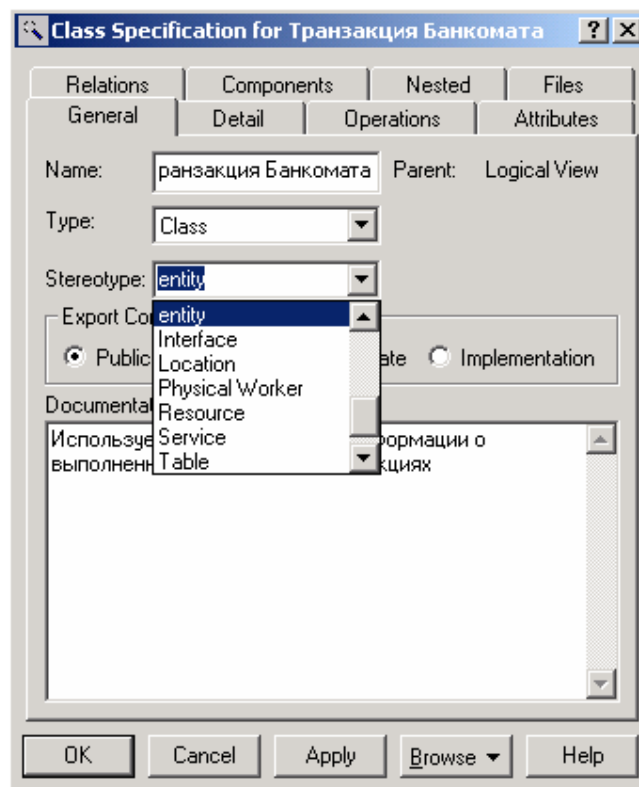


Рис. 4.2. Диалоговое окно спецификации свойств класса Транзакция Банкомата при выборе из вложенного списка стереотипа entity

Выбор данного стереотипа означает, что соответствующий *класс* предназначен для хранения информации, которая должна сохраняться в системе после уничтожения объектов данного *класса*. Далее в секцию документации данного *класса* можно ввести поясняющий текст: "Используется для сохранения информации о выполненных банкоматом транзакциях" и нажать кнопку **Apply** или **OK**, чтобы сохранить результаты редактирования свойств выбранного *класса*. После назначения стереотипа классу Транзакция банкомата текст данного стереотипа в угловых скобках появится выше имени данного *класса* (рис. 4.3).

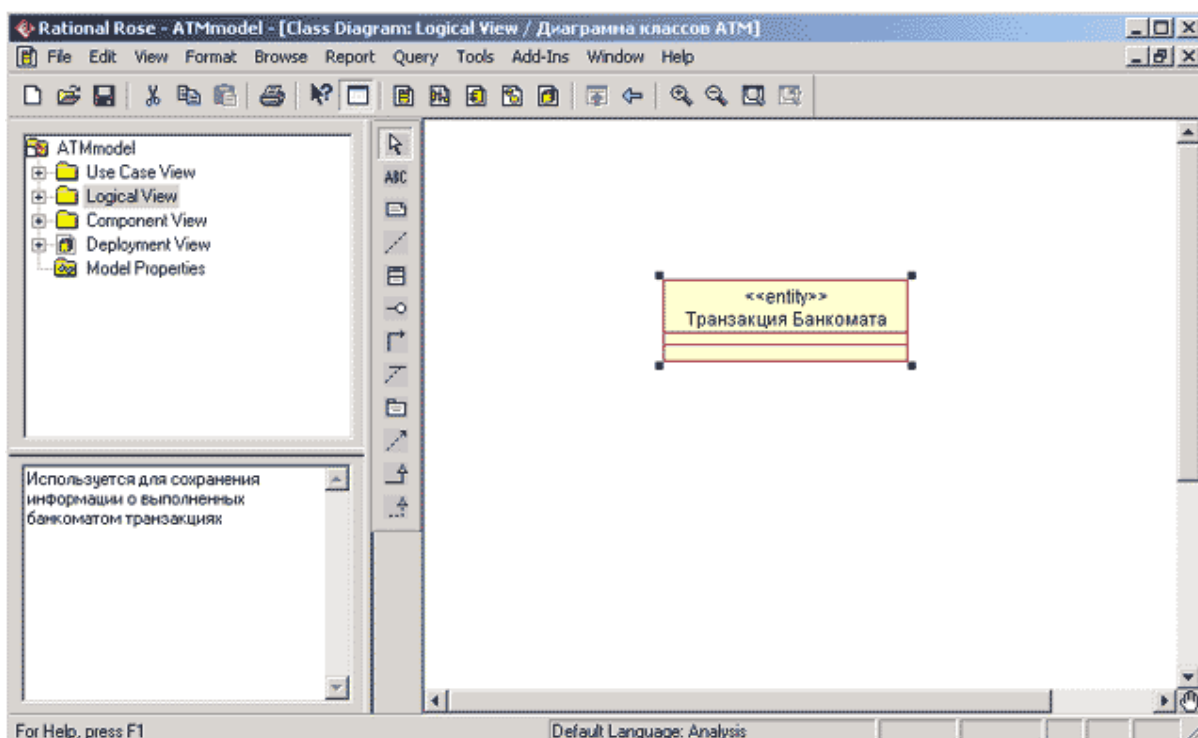


Рис. 4.3. Диаграмма классов модели банкомата после выбора стереотипа для класса Транзакция Банкомата

Для отдельного *класса* можно уточнить также и другие его свойства, доступные для редактирования на вкладке **Detail** (Подробно) окна спецификации свойств этого *класса*. Например, на этой вкладке с помощью вложенного списка **Multiplicity** (Кратность) можно задать количество объектов или экземпляров данного *класса*, для чего следует выбрать строку с буквой *n*. Данное значение означает, что у *класса* Транзакция банкомата может быть любое конечное число экземпляров (рис. 4.4). Поле ввода с именем **Space** (Пространство) служит для указания объема абсолютной или относительной памяти, которая требуется, по оценке разработчика, для реализации каждого объекта данного *класса*. Применительно к рассматриваемой модели это поле можно оставить пустым.

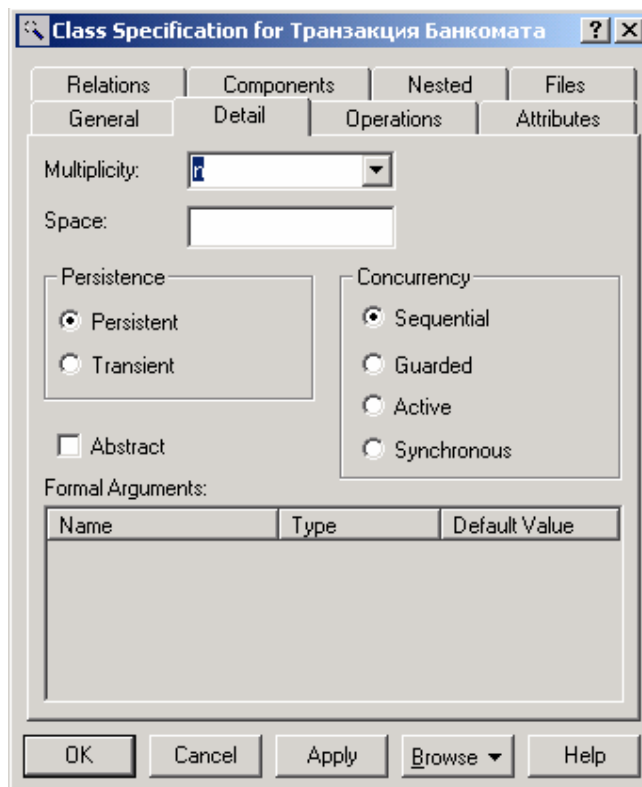


Рис. 4.4. Диалоговое окно спецификации свойств класса Транзакция Банкомата, открытое на вкладке **Detail** (Подробно)

Далее можно задать устойчивость *классов* в группе выбора **Persistence**. При этом выбор свойства **Persistent** (Устойчивый) означает, что информация об объектах данного *класса* должна быть сохранена в системе. Выбор свойства **Transient** (Временный) означает, что нет необходимости сохранять информацию об объектах данного *класса* в системе после завершения работы программного приложения. Применительно к рассматриваемой модели следует выбрать свойство **Persistent**.

В группе выбора **Concurrency** (Параллельность) можно специфицировать условия на возможность реализации объектов данного *класса* в параллельных потоках управления. Для выбора могут быть использованы следующие свойства:

- **Sequential** (Последовательный) - свойство по умолчанию, которое означает, что объекты *класса* будут вести себя нормально только при наличии одного потока управления, т. е. соответствующие операции объектов должны выполняться последовательно. В то же время при наличии нескольких потоков управления стабильное поведение объектов *класса* не гарантируется.
- **Guarded** (Безопасный) - означает, что при наличии нескольких потоков управления объекты *класса* будут вести себя ожидаемым от них образом. Для этого объекты в различных потоках должны взаимодействовать друг с другом для того, чтобы гарантировать отсутствие конфликта между ними.
- **Active** (Активный) - означает, что *класс* должен иметь свой собственный поток управления.
- **Synchronous** (Синхронный) - означает, что объекты *класса* будут вести себя ожидаемым от них образом при наличии нескольких потоков управления. При этом нет необходимости во взаимодействии объектов в различных потоках управления, поскольку объекты данного *класса* могут самостоятельно разрешать возможные конфликты.

Для того, чтобы специфицировать *класс* как абстрактный, т.е. не имеющий экземпляров, следует на этой же вкладке выставить отметку в свойстве **Abstract** (Абстрактный).

Применительно к рассматриваемой модели для *класса* Транзакция банкомата следует выбрать свойства **Persistent** и **Sequential**, а отметку для свойства **Abstract** оставить пустой.

Следует заметить, что для предотвращения потери информации о разрабатываемой модели и результатов редактирования свойств ее графических элементов необходимо периодически сохранять модель во внешнем файле. Для этого следует выполнить операцию главного меню: **File**→**Save** (Файл→Сохранить) или нажать комбинацию клавиш: **Ctrl+S**. Для этой же цели служит соответствующая кнопка на стандартной панели инструментов (см. табл. 1.1).

Стереотипы классов и их графическое представление

На разрабатываемой диаграмме *классов* выбран текстовый способ изображения стереотипов *классов*, при котором стереотип записывается в угловых кавычках выше имени соответствующего *класса*. Программа IBM Rational Rose 2003 позволяет альтернативно представлять стереотипы в форме специальных графических изображений (как в браузере проекта) или в форме небольших декоративных значков в верхней секции прямоугольника *класса* на диаграмме, а также вообще отказаться от изображения стереотипов.

Изменить изображение стереотипа для отдельного *класса* можно, например, с помощью одной из вложенных операций контекстного меню: **Options**→**Stereotype Display** (Параметры→Изображение стереотипа). В качестве примера можно представить изображение *класса* Транзакция Банкомата в форме специальной графической пиктограммы стереотипа. С этой целью следует выполнить операцию контекстного меню: **Options**→**Stereotype Display**→**Icon** (Параметры→Изображение стереотипа→Пиктограмма). Соответствующее графическое изображение стереотипа <<entity>> для *класса* Транзакция Банкомата в форме пиктограммы будет иметь следующий вид (рис. 4.5, а).

Для сравнения можно выбрать изображение *класса* Транзакция Банкомата в форме декоративного графического стереотипа. С этой целью выполним операцию контекстного меню: **Options**→**Stereotype Display**→**Decoration** (Параметры→Изображение стереотипа→Декорация). Соответствующее графическое изображение стереотипа <<entity>> для *класса* Транзакция Банкомата в форме декорации будет иметь следующий вид (рис. 4.5, б).

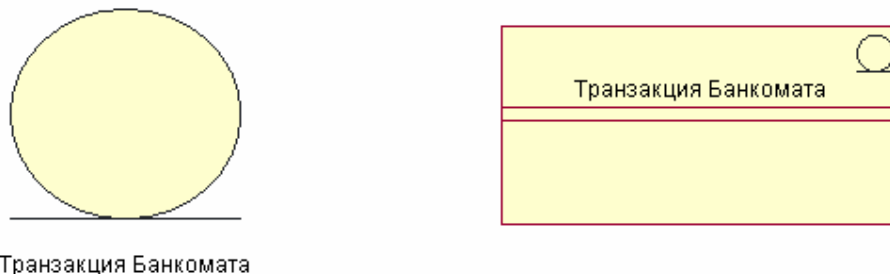


Рис. 4.5. Графические способы изображения стереотипа <<entity>> для класса Транзакция Банкомата

Изменить изображение стереотипов одновременно для нескольких *классов* диаграммы можно с помощью одной из вложенных операций главного меню: **Format**→**Stereotype Display** (Формат→Изображение стереотипов). В этом случае необходимо выделить все классы модели в окне диаграммы *классов* или в браузере проекта. Для выделения группы *классов* на диаграмме или в браузере проекта следует, удерживая нажатой клавишу **Ctrl**

или **Shift** на клавиатуре, последовательно щелкать на их изображении левой кнопкой мыши.

Выделить все графические элементы на диаграмме *классов*, также как и на любой другой диаграмме модели, можно с помощью выполнения операции главного меню: **Edit**→**Select All** (Редактирование→Выделить все) или с помощью комбинации клавиш **Ctrl+A**. Следует отметить, что выбор того или иного способа изображения стереотипов *классов* на диаграмме *классов* определяется разработчиком исходя из его личных предпочтений, и не оказывает влияния на содержательный аспект логического представления модели.

Продолжая разработку модели банкомата, добавим на диаграмму второй *класс* с именем **Контроллер Банкомата**, для которого в окне спецификации свойств выберем стереотип **control** (*управляющий класс*), а в качестве документации введем текст: "Реализует логику функционирования банкомата". При этом атрибуты и операции у данного *класса* будут отсутствовать. Соответствующий фрагмент диаграммы *классов* после добавления *управляющего класса* **Контроллер Банкомата** будет иметь следующий вид (рис. 4.6).

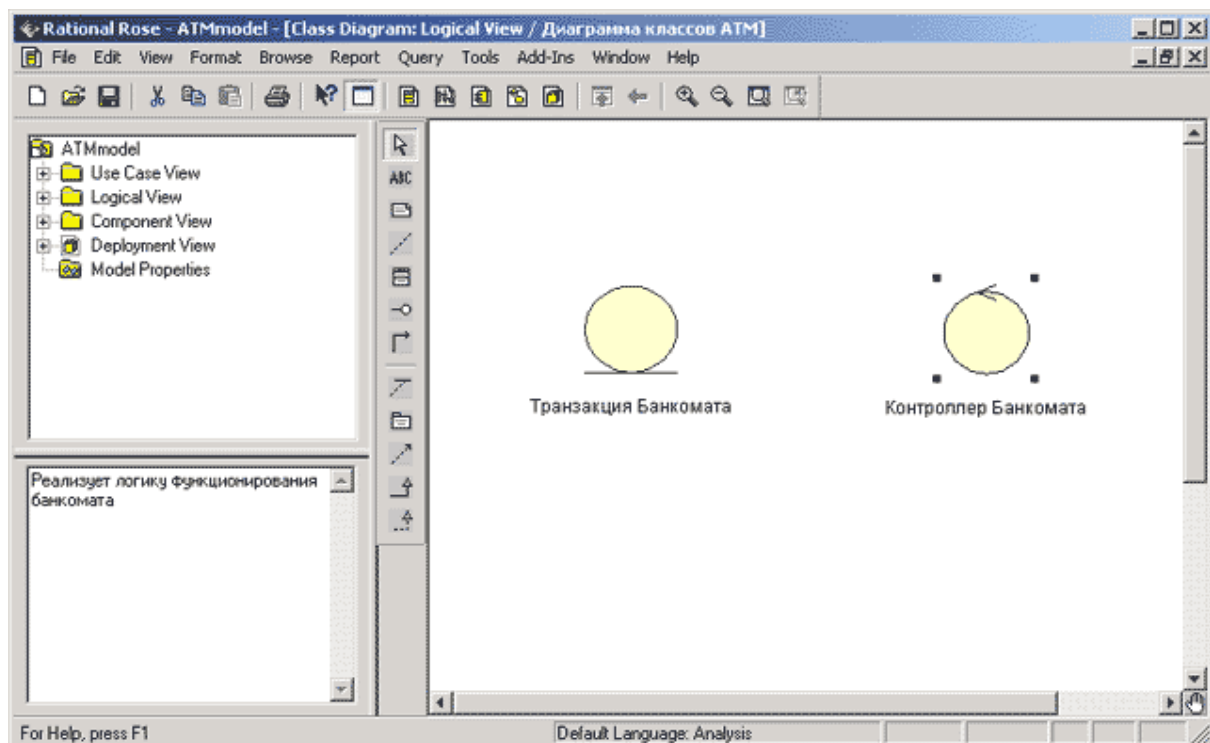


Рис. 4.6. Фрагмент диаграммы классов модели банкомата после добавления на нее класса **Контроллер Банкомата**

Продолжая разработку модели банкомата, добавим на диаграмму третий *класс* с именем **Устройство чтения карточки**, для которого в окне спецификации свойств выберем стереотип **boundary** (*граничный класс*). Применение этого стереотипа означает, что данный *класс* находится на границе моделируемой системы, в качестве которой рассматривается модель банкомата. После этого в секцию документации данного *класса* можно ввести поясняющий текст: "Устанавливается на банкомате".

Далее следует добавить *класс* с именем **IKонтроллер Банка**, для которого выбрать стереотип **Interface** (*Интерфейс*), означающий, что банкомат пользуется услугами Банка при обработке своих транзакций. Заметим, что первой буквой в имени этого *класса* является английское "I", которое служит в языке UML для указания *интерфейса*.

Соответствующий фрагмент диаграммы *классов* после добавления на нее *классов* Устройство чтения карточки и *ИКонтроллер* Банка будет иметь следующий вид (рис. 4.7).

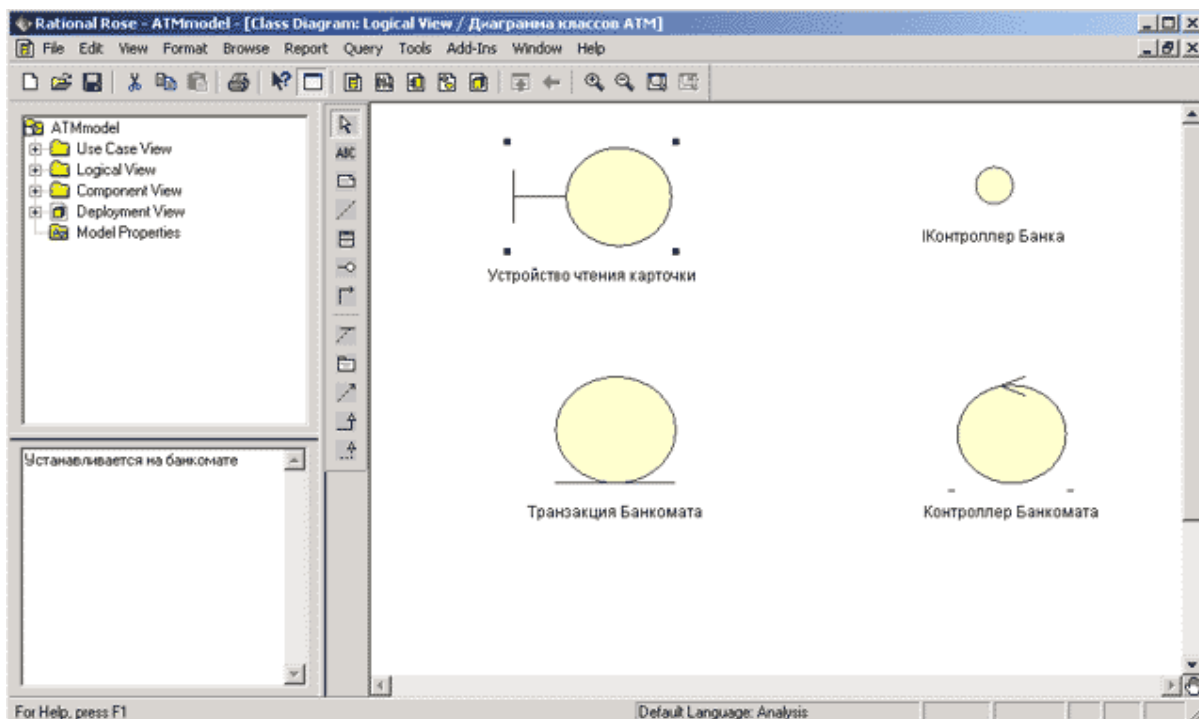


Рис. 4.7. Фрагмент диаграммы классов модели банкомата после добавления на нее классов Устройство чтения карточки и Контролер Банкомата

Практические действия по добавлению атрибутов и операций *классов*, редактированию их свойств, по добавлению отношений в процессе продолжения разработки диаграммы *классов* для модели банкомата, а также окончательный вид диаграммы *классов* для разрабатываемой модели банкомата будут рассмотрены в следующих лекциях.

5. Добавление атрибутов и операций на диаграмму классов

Добавление атрибутов классов. Диалоговое окно редактирования свойств атрибута класса. Добавление операций классов. Диалоговое окно редактирования свойств операции класса.

Добавление и редактирование атрибутов классов

Из всех графических элементов среды IBM Rational Rose 2003 класс обладает максимальным набором свойств, главными из которых являются его *атрибуты* и *операции*. Поскольку именно диаграмма классов используется в среде IBM Rational Rose 2003 для генерации программного кода, в данной лекции будут подробно рассмотрены соответствующие свойства *атрибутов* и *операций*.

Добавить *атрибут* к созданному ранее классу можно одним из следующих способов:

- С помощью *операции* контекстного меню **New Attribute** (Новый *атрибут*) для класса, выделенного на диаграмме классов. В этом случае активизируется курсор ввода текста в области графического изображения класса на диаграмме.
- С помощью *операции* контекстного меню: **New→Attribute** (Новый→*Атрибут*) для класса, выделенного в браузере проекта. В этом случае активизируется курсор

ввода текста в области иерархического представления класса в браузере проекта под именем соответствующего класса.

- С помощью *операции* контекстного меню **Insert** (Вставить), вызванного при позиционировании курсора в области открытой вкладки *атрибутов* в диалоговом окне свойств **Class Specification** соответствующего класса.

После добавления *атрибута* к классу по умолчанию ему присваивается имя `name` и некоторый *квантор видимости* (рис. 5.1).

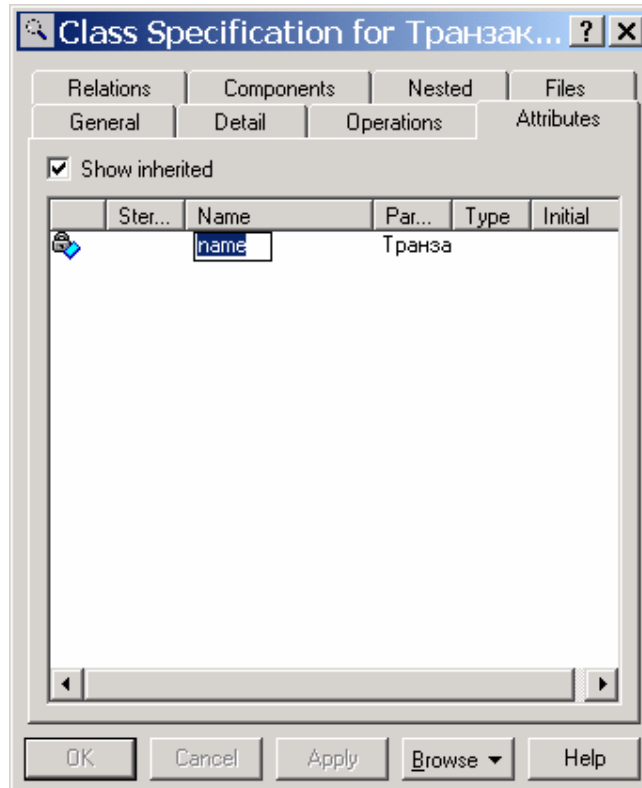


Рис. 5.1. Диалоговое окно спецификации свойств класса после добавления нового атрибута

Для рассматриваемой модели банкомата имя добавленного *атрибута* следует изменить на *идентификатор карточки*. Напомним, что имена *атрибутов* и *операций* классов должны начинаться со строчной буквы. Видимость *атрибутов* на диаграмме классов изображается в форме специальных пиктограмм или украшений. Используемые пиктограммы видимости изображаются перед именем соответствующего *атрибута* и имеют следующий смысл (табл. 5.1).

Таблица 5.1. Пиктограммы видимости атрибутов классов		
Графическое изображение	Текстовый аналог	Назначение пиктограммы
	Public	Общедоступный или открытый. В нотации языка UML такому <i>атрибуту</i> соответствует знак «+»
	Protected	Защищенный. В нотации языка UML такому <i>атрибуту</i> соответствует знак «#»
	Private	Закрытый. В нотации языка UML такому <i>атрибуту</i> соответствует знак «-»
	Implementation	Реализация. В нотации языка UML такому <i>атрибуту</i> соответствует знак «□»

Для редактирования свойств *атрибутов* предназначено специальное диалоговое окно спецификации *атрибута* **Class Attribute Specification**, которое открывается двойным щелчком мыши на строке выбранного *атрибута* в окне спецификации свойств класса. В окне свойств отдельного *атрибута* класса можно задать *тип* данных *атрибута* и его начальное *значение*, а также назначить *атрибуту* стереотип из раскрывающегося списка или изменить его *квантор* *видимости*.

Для *атрибута* идентификатор карточки в качестве *типа* его *допустимых значений* из вложенного списка **Type** следует выбрать тип **Integer** (целочисленный), а для задания *квантора* *видимости* следует выбрать в группе **Export Control** (Управление экспортом) *квантор* **Public**. Поскольку начальное *значение* для данного *атрибута* не определено, соответствующее поле ввода следует оставить пустым. В секцию документации данного *атрибута* класса можно ввести поясняющий текст: «Устройство чтения карточки считывает значение этого *атрибута* с кредитной карточки клиента» и нажать кнопку **Apply** или **OK**, чтобы сохранить результаты редактирования этих свойств *атрибута*. Соответствующее окно спецификации свойств *атрибута* идентификатор карточки после редактирования его общих свойств будет иметь следующий вид (рис. 5.2).

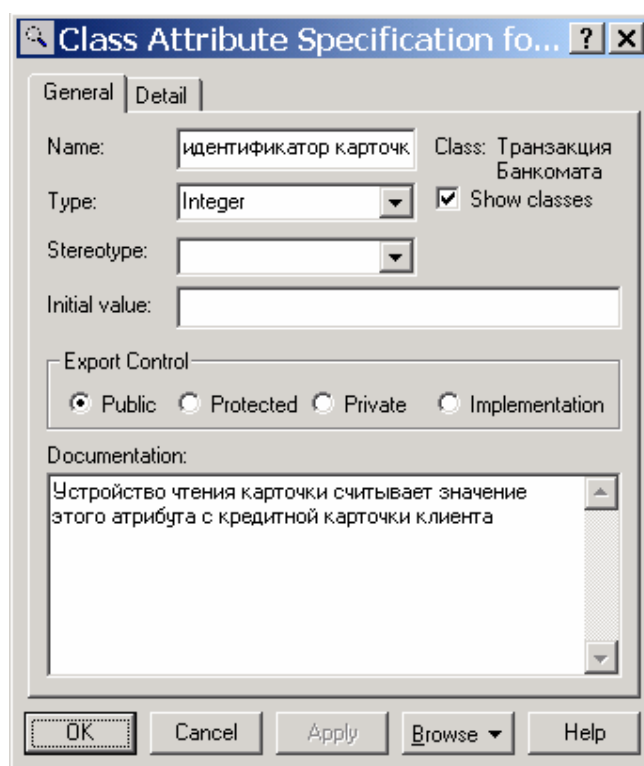


Рис. 5.2. Диалоговое окно спецификации свойств атрибута идентификатор карточки после его редактирования

Для отдельного *атрибута* можно также определить дополнительные свойства, доступные для редактирования на вкладке **Detail** (Подробно) диалогового окна спецификации свойств выбранного *атрибута* (рис. 5.3).

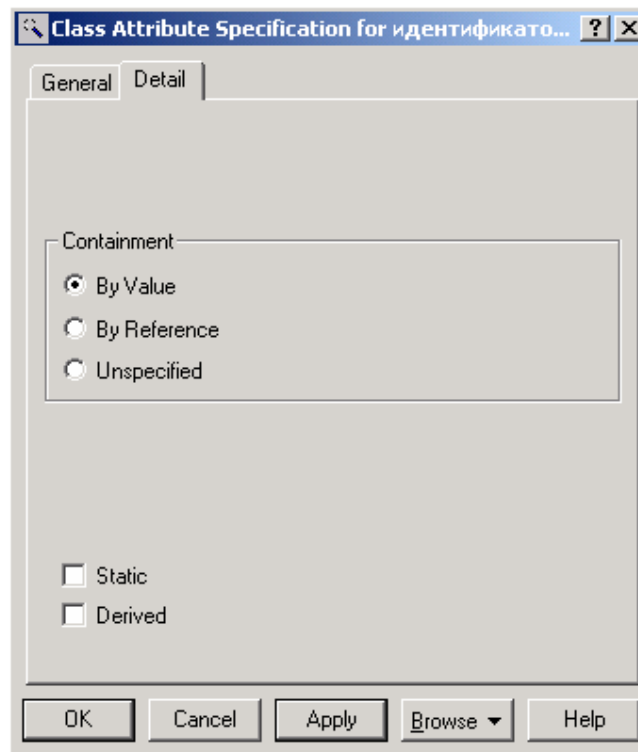


Рис. 5.3. Диалоговое окно спецификации свойств атрибута идентификатор карточки, открытое на вкладке **Detail** (Подробно)

На вкладке **Detail** в группе выбора **Containment** (Локализация) можно специфицировать условия хранения *атрибута* у объектов выбранного класса. Для выбора могут быть использованы следующие свойства:

- **By value** (По значению) - свойство по умолчанию, которое означает, что значения *атрибута* хранятся в пределах адресного пространства, выделенного для объекта данного класса. Например, если имеется атрибут типа **String**, то значение этой строки содержится в пределах определения класса.
- **By reference** (По ссылке) - означает, что значение *атрибута* хранится вне адресного пространства, выделенного для объекта данного класса, но у объектов класса имеется указатель на этот атрибут.
- **Unspecified** (Не определен) - означает, что метод локализации данного *атрибута* не определен. В этом случае при генерации программного кода для данного *атрибута* по умолчанию выбирается значение **By value**.

Далее можно определить атрибут как статичный, выставив отметку в строке выбора **Static**. Статичный атрибут по определению имеет одно и тоже значение для всех объектов рассматриваемого класса. Наконец, на вкладке **Detail** можно определить атрибут как производный, выставив отметку в строке выбора **Derived**. Значение производного *атрибута* по определению может быть вычислено на основании значений других *атрибутов* этого или другого класса.





Добавление и редактирование операций классов

Функционирование банкомата основано на выполнении отдельными его устройствами тех или иных действий. В модели структуры банкомата все действия представляются с помощью *операций* классов. Таким образом, следующий этап разработки диаграммы классов связан со спецификацией *операций* классов.

Добавить *операцию* к созданному ранее классу можно одним из следующих способов:

- С помощью *операции* контекстного меню **New Operation** (Новая операция) для класса, выделенного на диаграмме классов. В этом случае активизируется курсор ввода в области графического изображения класса на диаграмме.
- С помощью *операции* контекстного меню: **New→Operation** (Новая→Операция) для класса, выделенного в браузере проекта. В этом случае активизируется курсор ввода в области иерархического представления класса в браузере под именем соответствующего класса.
- С помощью *операции* контекстного меню **Insert** (Вставить), вызванного при позиционировании курсора в области открытой вкладки *операций* в диалоговом окне свойств **Class Specification** соответствующего класса.

После добавления *операции* к классу по умолчанию ей присваивается имя **opname** и некоторый *квантор видимости*. Видимость *операций* на диаграмме классов также изображается в форме специальных пиктограмм или украшений. Используемые пиктограммы видимости изображаются перед именем соответствующей *операции* и имеют следующий смысл (табл. 5.2).

Таблица 5.2. Пиктограммы видимости операций классов		
Графическое изображение	Текстовый аналог	Назначение пиктограммы
	Public	Общедоступный или открытый. В нотации языка UML такому <i>атрибуту</i> соответствует знак «+»
	Protected	Защищенный. В нотации языка UML такому <i>атрибуту</i> соответствует знак «#»
	Private	Закрытый. В нотации языка UML такому <i>атрибуту</i> соответствует знак «-»
	Implementation	Реализация. В нотации языка UML такому <i>атрибуту</i> соответствует знак «□»

В контексте рассматриваемой модели банкомата в качестве имени первой *операции* для класса **Транзакция Банкомата** следует задать: **создать новую транзакцию**. При этом скобки при задании имени *операции* не записываются, поскольку программа IBM Rational Rose 2003 добавляет их автоматически. Однако, следуя правилам именования *операций* в языке UML, в тексте лекции имена *операций* будут указываться со скобками.

Каждая из *операций* классов имеет собственное диалоговое окно спецификации свойств **Operation Specification**, которое может быть открыто по двойному щелчку на имени *операции* на соответствующей вкладке спецификации класса или на имени этой *операции* в браузере проекта. Для *операции* **создать новую транзакцию()** в качестве *квантора видимости* следует выбрать из вложенного списка *квантор public*. В секцию документации данной *операции* класса можно ввести поясняющий текст: «Вызывается после того, как кредитная карточка вставлена в Устройство чтения карточки» и нажать кнопку **Apply** или **OK**, чтобы сохранить результаты редактирования свойств этой *операции*. Соответствующее окно спецификации свойств *операции* **создать новую транзакцию()** после редактирования ее свойств будет иметь следующий вид (рис. 5.4).

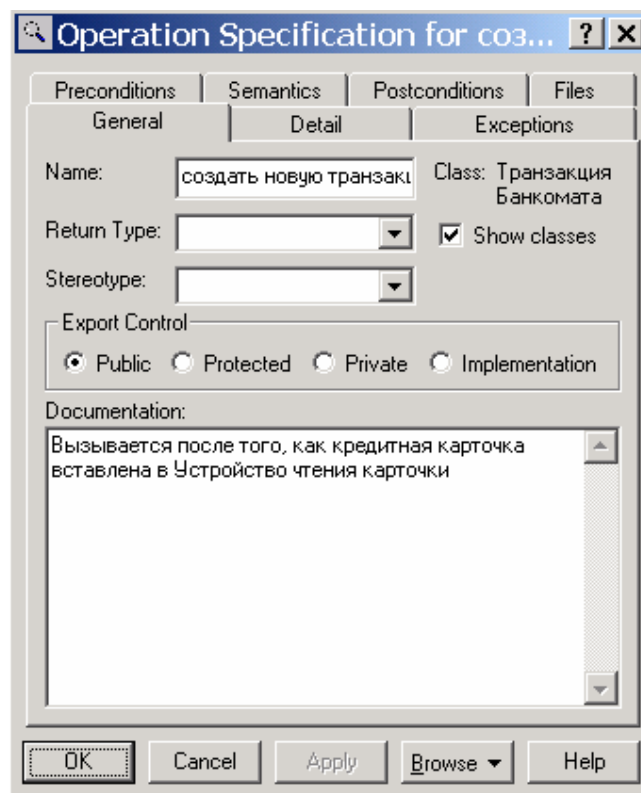


Рис. 5.4. Диалоговое окно спецификации свойств операции создать новую транзакцию ()

Для операций классов кроме *квантора видимости* можно также задать: *аргументы* и их тип, *тип возвращаемого результата*, *стереотип операции*, а также определить протокол и размер, задать исключительные ситуации, специфицировать предусловия и постусловия и целый ряд других свойств. Для отдельной *операции* эти дополнительные свойства доступны для редактирования на вкладке **Detail** (Подробно) диалогового окна спецификации свойств выбранной *операции* (рис. 5.5).

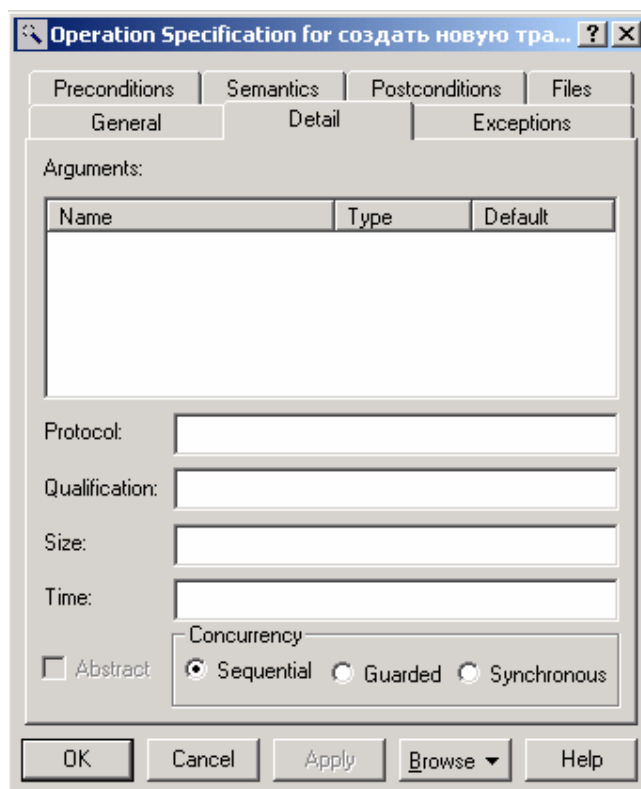


Рис. 5.5. Диалоговое окно спецификации свойств операции создать новую транзакцию(), открытое на вкладке **Detail** (Подробно)

На вкладке **Detail** в многостраничном поле **Arguments** (Аргументы) можно определить *аргументы редактируемой операции*. Для этого следует выполнить *операцию* контекстного меню **Insert** (Вставить). После этого в этом поле появится аргумент данной *операции* с именем по умолчанию **argname**. Для редактирования свойств аргумента предназначено специальное окно свойств аргумента.

На вкладке **Detail** в поле **Protocol** (Протокол) можно специфицировать порядок выполнения *операций* класса, например, указать, что одна *операция* не может быть вызвана раньше другой. Соответствующий текст в данное поле вводится с клавиатуры и попадает в генерируемый код в форме комментария. В поле **Qualification** (Квалификация) можно уточнить детали реализации *операции*, связанные с конкретным языком программирования. Соответствующий текст также вводится в данное поле с клавиатуры и попадает в генерируемый код в форме комментария.

Далее на этой же вкладке в полях **Size** (Размер) и **Time** (Время) можно специфицировать предполагаемый объем памяти и время, необходимое для выполнения *операции*. Соответствующая информация попадает в генерируемый код в форме комментария.

В группе выбора **Concurrency** (Параллельность) можно специфицировать условия на возможность параллельного выполнения данной *операции*. Для выбора могут быть использованы следующие свойства:

- **Sequential** (Последовательная) - свойство по умолчанию, которое означает, что данная *операция* класса может быть выполнена только при наличии одного потока управления, т. е. соответствующая *операция* класса должна выполняться последовательно. При наличии нескольких потоков управления выполнение данной *операции* класса не гарантируется.
- **Guarded** (Безопасная) - означает, что при наличии нескольких потоков управления выполнение данной *операции* класса гарантируется только в том

случае, когда обеспечено взаимодействие объектов друг с другом в различных потоках.

- **Synchronous** (Синхронная) - означает, что выполнение данной *операции* класса гарантируется при наличии нескольких потоков управления. При этом нет необходимости во взаимодействии объектов в различных потоках управления, поскольку данная *операция* класса будет выполняться в отдельном потоке управления вплоть до своего завершения.

Применительно к рассматриваемой модели для *операции* `создать новую транзакцию()` следует выбрать свойство **Sequential**, а поля всех других свойств оставить пустыми.

Спецификация атрибутов и операций для класса Транзакция Банкомата

Чтобы закончить спецификацию класса Транзакция Банкомата аналогичным способом следует добавить еще 3 *атрибута* и 2 *операции* со следующими свойствами:

- *значение ПИН-кода карточки* с *квантором видимости* **public**. В качестве типа этого *атрибута* следует выбрать тип **Integer** (целочисленный), а в секцию документации *атрибута* ввести поясняющий текст: «Устройство чтения карточки считывает значение этого *атрибута* с кредитной карточки клиента».
- *введенный ПИН-код* с *квантором видимости* **public**. В качестве типа этого *атрибута* следует выбрать тип **Integer** (целочисленный), а в секцию документации *атрибута* ввести поясняющий текст: «Значение этого *атрибута* вводится клиентом с клавиатуры банкомата».
- *введенная сумма наличных* с *квантором видимости* **public**. В качестве типа этого *атрибута* следует выбрать тип **Currency** (Денежный), а в секцию документации *атрибута* ввести поясняющий текст: «Значение этого *атрибута* вводится клиентом с клавиатуры банкомата».
- *проверить правильность ПИН-кода()* с *квантором видимости* **public**. В качестве типа *возвращаемого результата* для этой *операции* следует выбрать тип **Boolean** (логический), а в секцию ее документации ввести поясняющий текст: «Вызывается после того, как клиент ввел значение ПИН-кода с клавиатуры банкомата».
- *завершить транзакцию()* с *квантором видимости* **public**. В секцию ее документации ввести поясняющий текст: «Вызывается после завершения всех действий банкомата по обслуживанию клиента».

Выполнить эти действия предлагается читателям самостоятельно. Соответствующий фрагмент диаграммы классов после добавления и спецификации *атрибутов* и *операций* для класса *Транзакция Банкомата* будет иметь следующий вид (рис. 5.6).

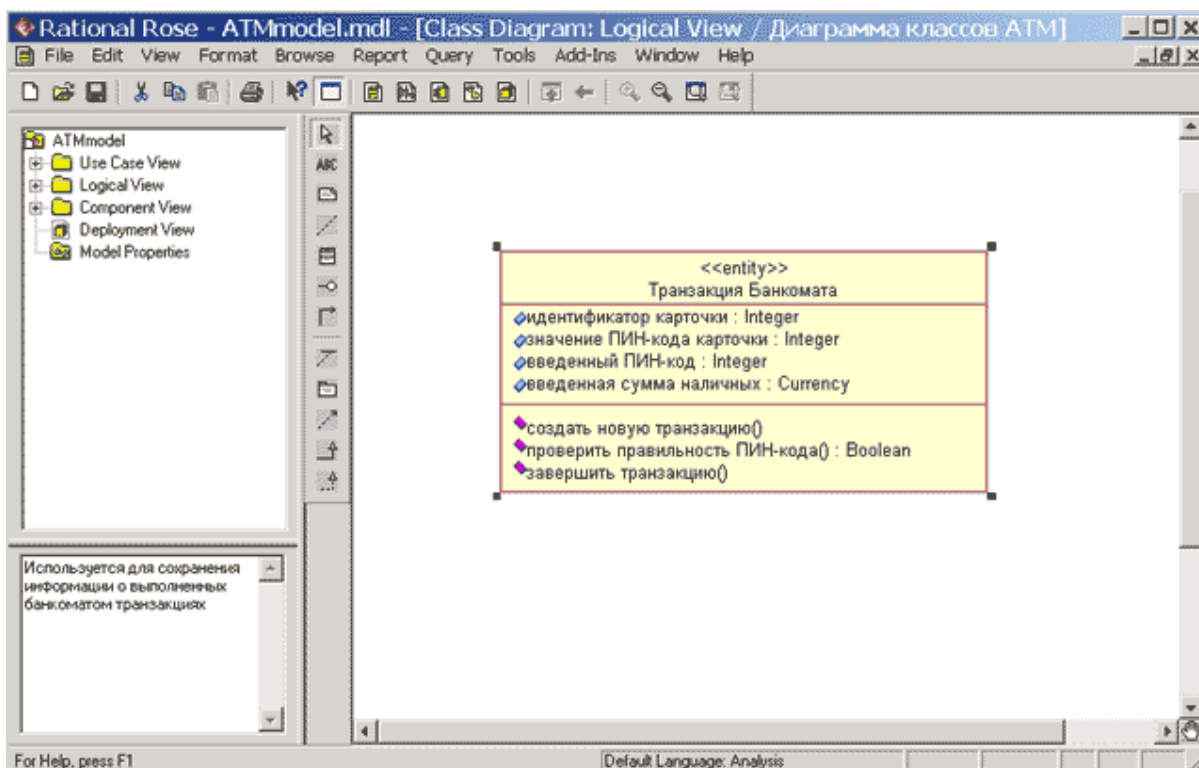


Рис. 5.6. Фрагмент диаграммы классов модели банкомата после добавления атрибутов и операций для класса Транзакция банкомата

Практические действия по добавлению *атрибутов* и *операций* для других классов, а также построение окончательной диаграммы классов для разрабатываемой модели банкомата будут рассмотрены в следующей лекции.

6. Добавление отношений на диаграмму классов и редактирование их свойств

Добавление ассоциации на диаграмму классов и редактирование ее свойств. Добавление отношений агрегации и композиции и редактирование их свойств. Добавление отношения обобщения и редактирование его свойств. Окончательное построение диаграммы классов модели банкомата.

Диаграмма классов является логическим представлением структуры модели, поэтому она должна содержать столько классов, сколько необходимо для реализации всего проекта. При этом для полного представления структуры модели необходимо установить и специфицировать отношения между классами. Этой теме и посвящена настоящая лекция.

Добавление ассоциации на диаграмму классов и редактирование ее свойств

Добавление на диаграмму *ассоциации* между двумя классами выполняется следующим образом. На специальной панели инструментов необходимо нажать кнопку с изображением пиктограммы направленной *ассоциации* и отпустить левую кнопку мыши. Если *ассоциация* - направленная, то на диаграмме классов надо выделить первый элемент *ассоциации* или источник, от которого исходит стрелка, и, не отпуская нажатую левую кнопку мыши, переместить ее указатель ко второму элементу отношения или приемнику, к которому направлена стрелка. После перемещения ко второму элементу кнопку мыши

следует отпустить, в результате чего на диаграмму классов будет добавлена направленная ассоциация с именем **Untitled** между двумя выбранными классами.

Продолжая разработку диаграммы классов модели банкомата, добавим на нее описанным способом направленную ассоциацию между классом **Контроллер Банкомата** и классом **Транзакция Банкомата** (рис. 6.1).

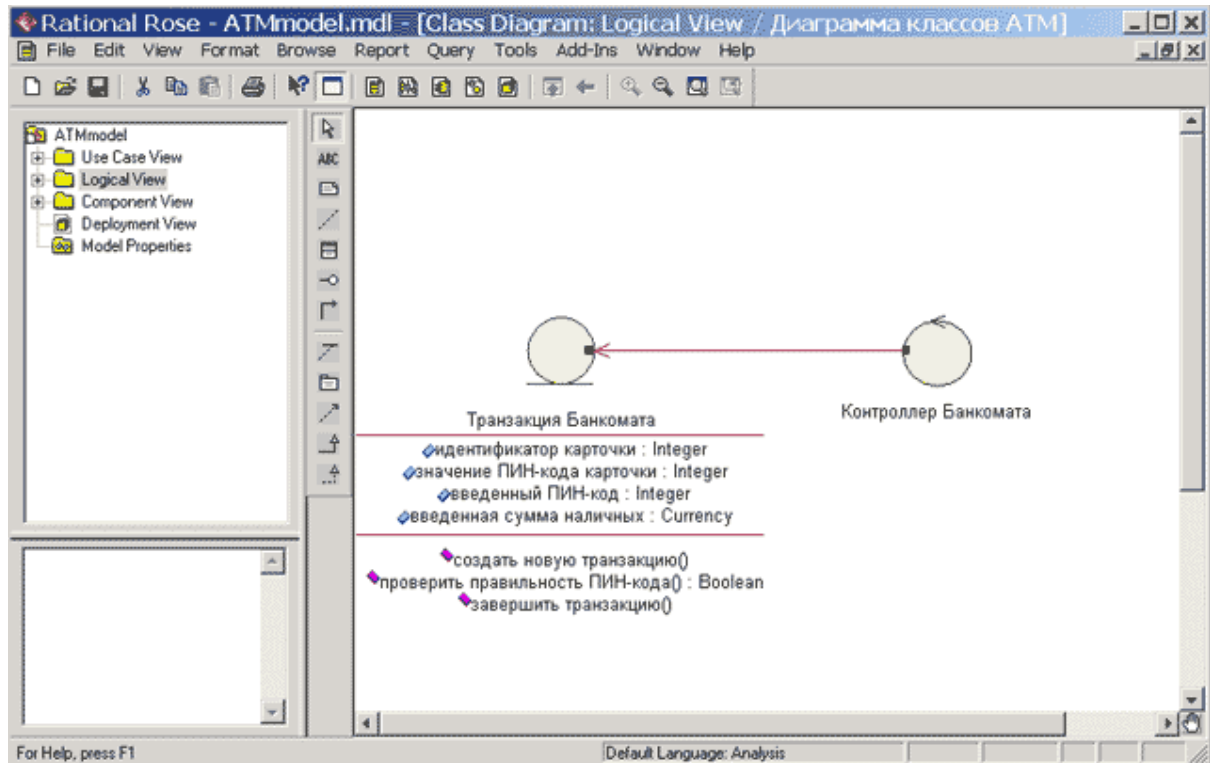


Рис. 6.1. Фрагмент диаграммы классов модели банкомата после добавления на неё направленной ассоциации

Изменим имя для данной ассоциации, предложенное средой по умолчанию. Это можно выполнить с помощью окна спецификации свойств ассоциации. Доступ к диалоговому окну спецификации свойств ассоциации **Association Specification** можно получить после выделения линии ассоциации на диаграмме классов или в браузере проекта и двойного щелчка на ней левой кнопки мыши (рис. 6.2).

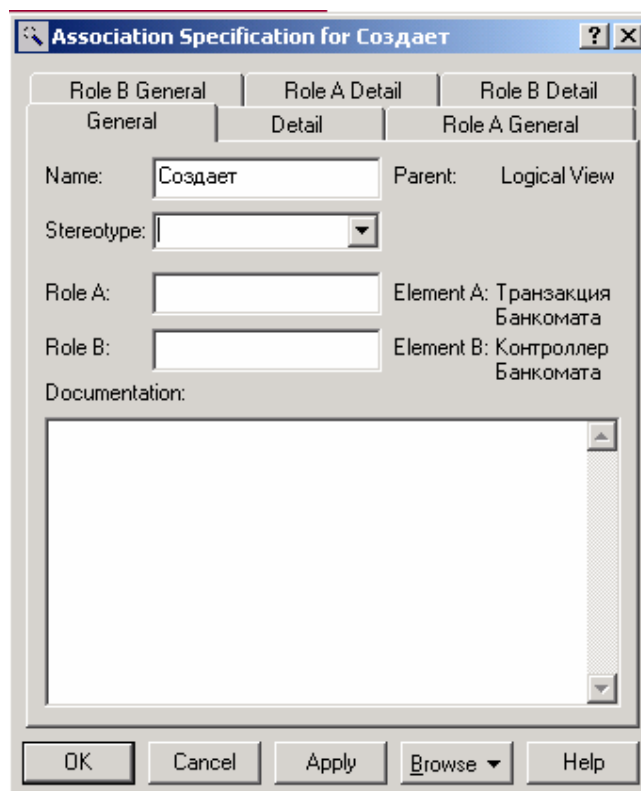


Рис. 6.2. Диалоговое окно спецификации свойств ассоциации

Для задания имени *ассоциации* следует на вкладке **General** (Общие) в поле ввода **Name** (Имя) ввести текст ее имени: Создает и нажать кнопку **Apply** или **OK**, чтобы сохранить результаты редактирования имени *ассоциации*. Для *ассоциации* можно задать также *кратность* каждого из концов *ассоциации*, стереотип, использовать ограничения и роли, а также некоторые другие свойства.

Для добавленной на диаграмму классов *ассоциации* зададим *кратность* конца *ассоциации* у класса *Контроллер Банкомата*, равную 1. Для этого следует в окне спецификации свойств *ассоциации* перейти на вкладку **Role B Detail** и выбрать значение 1 из вложенного списка **Multiplicity**. Аналогичным образом следует задать *кратность* конца *ассоциации* у класса *Транзакция Банкомата* равную 1..n, для чего на вкладке **Role A Detail** и следует выбрать значение 1..n из вложенного списка **Multiplicity**. Содержательно это будет означать, что каждый объект класса *Контроллер Банкомата* может быть связан с одним или несколькими объектами класса *Транзакция Банкомата*.

Если *ассоциация* является ненаправленной, то порядок выбора классов может быть произвольный, а после добавления *ассоциации* на диаграмму классов следует изменить значение соответствующего свойства данной *ассоциации*. С этой целью необходимо перейти на вкладку **Role A Detail** в окне спецификации свойств *ассоциации* и убрать отметку у свойства **Navigable** (Навигация).

Добавление отношений агрегации и композиции на диаграмму классов и редактирование их свойств

Добавить на диаграмму отношение *агрегации* между двумя классами можно следующими способами:

- Щелкнуть на кнопке с изображением отношения *агрегации* на специальной панели инструментов и провести линию *агрегации* от одного класса к другому.

- Провести линию *ассоциации* между выбранными классами и изменить ее свойства таким образом, чтобы превратить данную *ассоциацию* в *агрегацию*.

В первом случае может оказаться, что по умолчанию на специальной панели инструментов диаграммы классов отсутствует кнопка с пиктограммой *агрегации*. В этом случае необходимо предварительно добавить ее на панель инструментов одним из описанных ранее способов. Во втором случае следует открыть окно спецификации свойств *ассоциации* **Association Specification** и на вкладке деталей соответствующего конца *ассоциации* выставить отметку в строке выбора **Aggregate** (*Агрегация*).

В качестве примера изменим тип созданной ранее *ассоциации* и сделаем ее агрегацией. Содержательно это будет означать, что класс **Контроллер Банкомата** будет включать в себя в качестве составной части класс **Транзакция Банкомата**; при этом уничтожение любого объекта класса **Контроллер Банкомата** не должно привести к уничтожению ассоциированных с ним объектов класса **Транзакция Банкомата**. С этой целью на вкладке **Role B Detail** деталей конца *ассоциации* класса **Контроллер Банкомата** следует выставить отметку в строке выбора **Aggregate** (рис. 6.3).

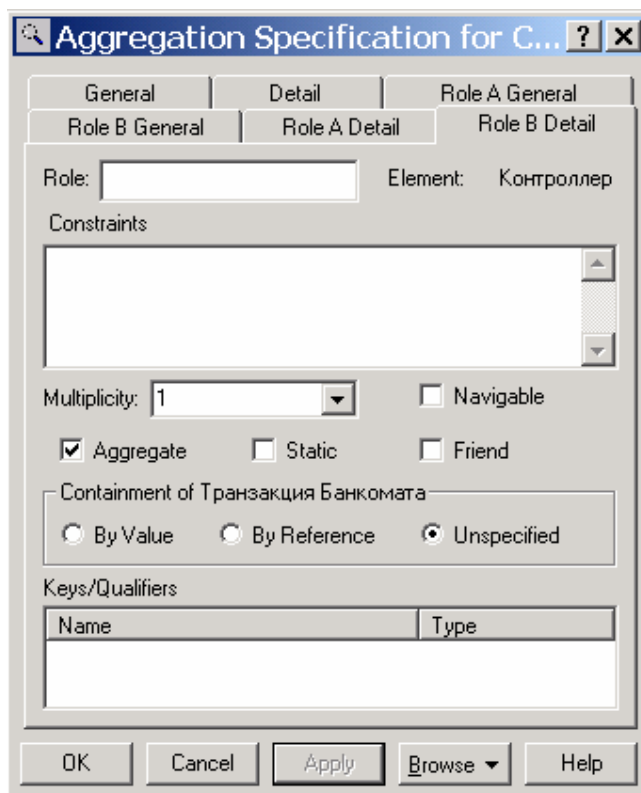


Рис. 6.3. Диалоговое окно спецификации свойств ассоциации

Соответствующий фрагмент диаграммы классов после изменения *ассоциации* между классами **Контроллер Банкомата** и **Транзакция Банкомата** на отношение *агрегации* будет иметь следующий вид (рис. 6.4).

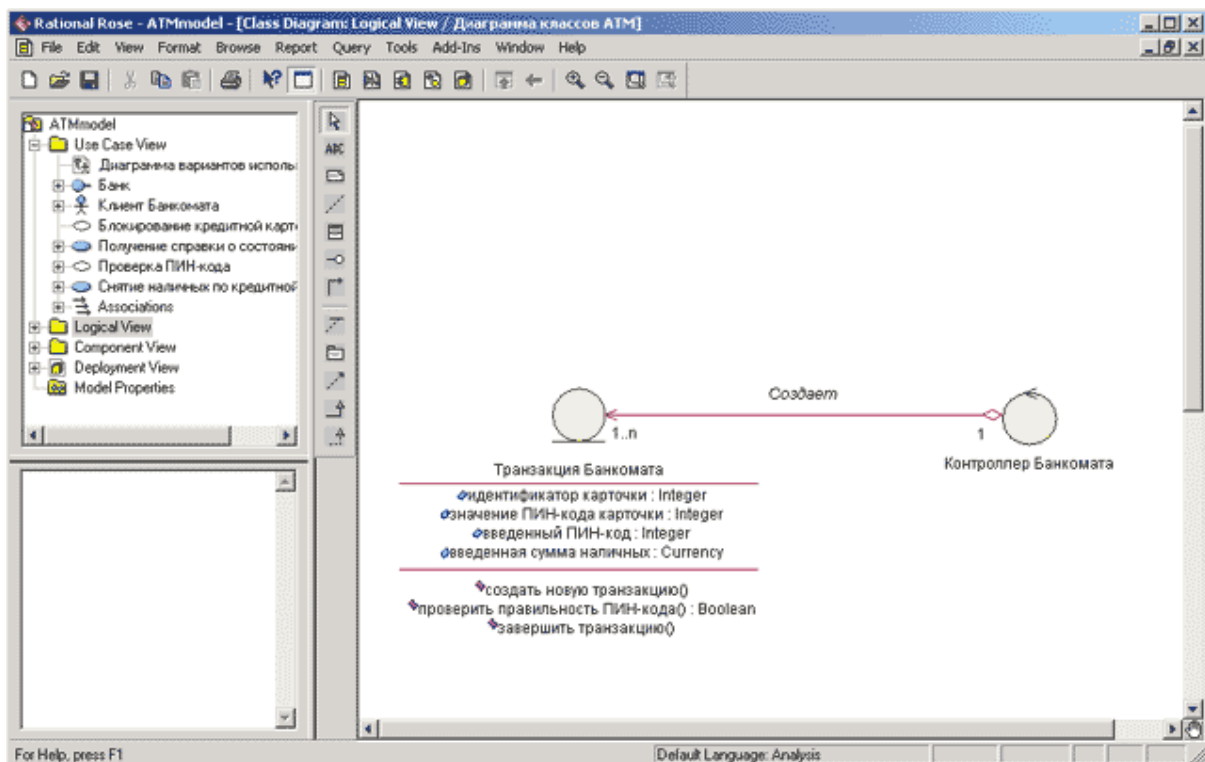


Рис. 6.4. Фрагмент диаграммы классов модели банкомата после добавления на нее отношения агрегации

Для изображения отношения *композиции* можно также вначале изобразить обычную ассоциацию, после чего, открыв окно ее свойств на вкладке деталей соответствующего конца ассоциации, (рис. 6.3) выставить отметку в строке выбора **Aggregate** (*Агрегация*) и в секции **Containment** (*Локализация*) выбрать опцию **By Value** (*По значению*). По умолчанию эта опция не специфицирована, т.е. выставлена отметка опции **Unspecified**.

Добавление отношения обобщения на диаграмму классов и редактирование ее свойств

Добавление на диаграмму отношения *обобщения* между двумя классами выполняется следующим образом. На специальной панели инструментов необходимо нажать кнопку с изображением пиктограммы *обобщения* и отпустить левую кнопку мыши. Далее на диаграмме классов надо выделить первый элемент *обобщения* или потомок, от которого исходит стрелка, и, не отпуская нажатую левую кнопку мыши, переместить ее указатель ко второму элементу отношения или предку, к которому направлена стрелка. После перемещения ко второму элементу кнопку мыши следует отпустить, в результате чего на диаграмму классов будет добавлена линия *обобщения* с именем **Untitled** между двумя выбранными классами.

Продолжая разработку диаграммы классов модели банкомата, добавим на нее описанным способом направленную *ассоциацию* между классом **Контроллер Банкомата** и дополнительно созданным абстрактным классом **Контроллер** (рис. 6.5). Последний класс может быть предназначен для спецификации системных атрибутов и операций, необходимых при исполнении соответствующей программы. Напомним, что на абстрактный характер класса указывает написание курсивом его имени, а для спецификации данного свойства класса необходимо на вкладке **Detail** (*Подробно*) окна спецификации свойств класса **Контроллер** выставить отметку в строке выбора **Abstract**.

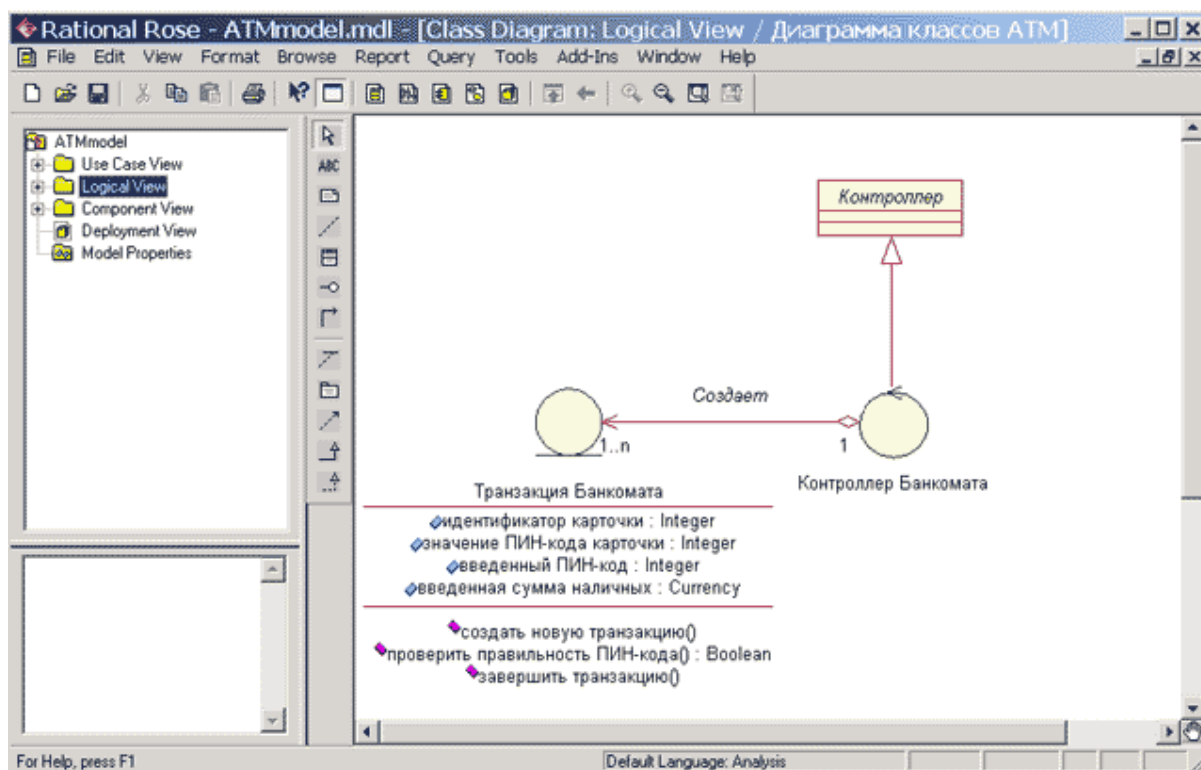


Рис. 6.5. Диаграмма классов модели банкомата после добавления на неё отношения обобщения

Изменим имя отношения *обобщения*, предложенное средой по умолчанию. Это можно выполнить с помощью окна спецификации свойств отношения *обобщения*. Доступ к диалоговому окну спецификации свойств отношения *обобщения* **Generalize Specification** можно получить после выделения линии *обобщения* на диаграмме классов или в браузере проекта и двойного щелчка на ней левой кнопки мыши (рис. 6.6).

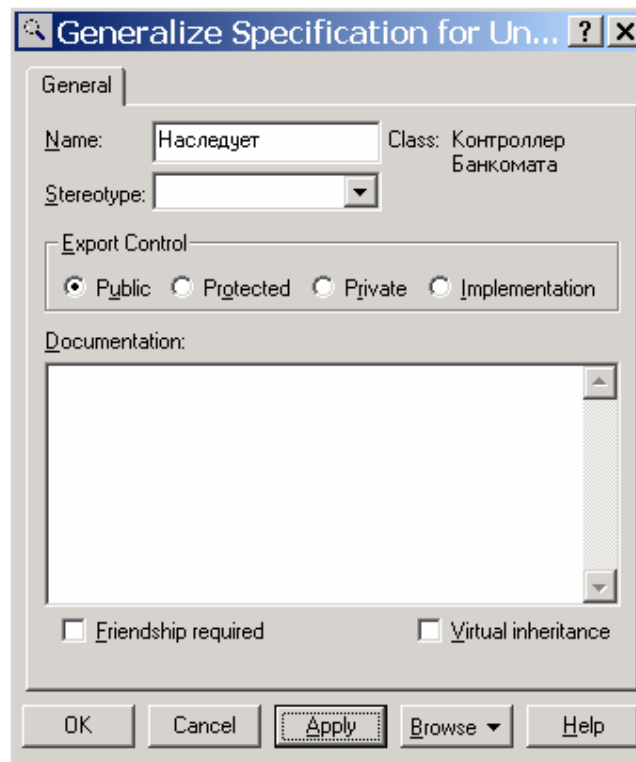


Рис. 6.6. Диалоговое окно спецификации свойств отношения обобщения

Для задания имени *обобщения* следует на единственной вкладке **General** (Общие) в поле ввода **Name** (Имя) ввести текст ее имени: *Наследует* и нажать кнопку **Apply** или **OK**, чтобы сохранить результаты редактирования имени *ассоциации*.

Окончательное построение диаграммы классов модели банкомата

Для окончательного построения диаграммы классов рассматриваемой модели банкомата следует описанным выше способом добавить оставшиеся классы и *ассоциации*, а также специфицировать стереотипы, атрибуты и операции этих классов. С этой целью следует выполнить следующие действия:

1. Для класса *Интерфейс Банка* добавить операцию: *проверить идентификатор карточки* (идентификатор карточки: *Integer*) с квантором видимости **public**. В качестве типа возвращаемого результата для этой операции следует выбрать тип **Boolean** (логический), а в качестве целочисленного аргумента задать *идентификатор карточки*. Для задания аргумента необходимо перейти на вкладку **Detail** (Подробно) окна спецификации свойств данной операции и после добавления аргумента с помощью операции контекстного меню **Insert** ввести имя аргумента и его тип **Integer** в соответствующие поля ввода.
2. Для класса *Интерфейс Банка* добавить операцию: *открыть счет клиента* (идентификатор карточки: *Integer*) с квантором видимости **public**. В качестве целочисленного аргумента этой операции следует задать *идентификатор карточки*.
3. Для класса *Интерфейс Банка* добавить операцию: *проверить баланс клиента* (идентификатор карточки: *Integer*, введенная сумма наличных: *Currency*) с квантором видимости **public**. В качестве типа возвращаемого результата для этой операции следует выбрать тип **Boolean** (логический). В качестве первого целочисленного аргумента этой операции следует задать *идентификатор карточки*, а в

- качестве второго аргумента - введенная сумма наличных с типом **Currency** (Денежный).
4. Для класса **Интерфейс Банка** добавить операцию: **уменьшить счет клиента** (идентификатор карточки: **Integer**, введенная сумма наличных: **Currency**) с квантором видимости **public**. В качестве типа возвращаемого результата для этой операции следует выбрать тип **Boolean** (логический). В качестве первого целочисленного аргумента этой операции следует задать идентификатор карточки, а в качестве второго аргумента - введенная сумма наличных с типом **Currency** (Денежный).
 5. Для класса **Устройство чтения карточки** добавить операцию: **прочитать идентификатор карточки()** с квантором видимости **public**. В качестве типа возвращаемого результата для этой операции следует выбрать тип **Integer** (целочисленный), а в секцию документации данной операции следует ввести поясняющий текст: «Вызывается после того, как кредитная карточка вставлена в Устройство чтения карточки».
 6. Для класса **Устройство чтения карточки** добавить операцию: **прочитать ПИН-код()** с квантором видимости **public**. В качестве типа возвращаемого результата для этой операции следует выбрать тип **Integer** (целочисленный), а в секцию документации данной операции следует ввести поясняющий текст: «Вызывается после того, как кредитная карточка вставлена в Устройство чтения карточки».
 7. Для класса **Устройство чтения карточки** добавить операцию: **вернуть кредитную карточку()** с квантором видимости **public**. В секцию документации данной операции следует ввести поясняющий текст: «Вызывается после завершения транзакции».
 8. Для класса **Устройство чтения карточки** добавить операцию: **блокировать кредитную карточку()** с квантором видимости **public**. В секцию документации данной операции следует ввести поясняющий текст: «Вызывается после того, как установлен факт утраты кредитной карточки владельцем».
 9. Добавить класс с именем **Экран Банкомата**, для которого выбрать стереотип **boundary**. Данный класс также находится на границе моделируемой системы, на что и указывает этот стереотип. В секцию документации данного класса следует ввести поясняющий текст: «Устанавливается на банкомате».
 10. Для класса **Экран Банкомата** добавить операцию: **показать меню опций()** с квантором видимости **public**.
 11. Для класса **Экран Банкомата** добавить операцию: **показать меню снятия суммы()** с квантором видимости **public**.
 12. Добавить класс с именем **Клавиатура Банкомата**, для которого выбрать стереотип **boundary**. В секцию документации данного класса следует ввести поясняющий текст: «Устанавливается на банкомате».
 13. Для класса **Клавиатура Банкомата** добавить операцию: **ввести ПИН-код()** с квантором видимости **public**. В качестве типа возвращаемого результата для этой операции следует выбрать тип **Integer**, а в секцию документации данной операции следует ввести поясняющий текст: «Вызывается после того, как клиент ввел значение ПИН-кода с клавиатуры».
 14. Для класса **Клавиатура Банкомата** добавить операцию: **ввести тип транзакции()** с квантором видимости **public**. В качестве типа возвращаемого результата для этой операции следует выбрать тип **Boolean** (логический), а в секцию документации данной операции следует ввести поясняющий текст: «Возвращает значение **Истина**, если клиент выбирает снятие наличных, и значение **Ложь**, если клиент выбирает получение справки о состоянии счета».
 15. Для класса **Клавиатура Банкомата** добавить операцию: **ввести сумму снятия наличных()** с квантором видимости **public**. В качестве типа возвращаемого результата для этой операции следует выбрать тип **Currency** (Денежный), а в секцию документации данной операции следует ввести поясняющий текст: «Вызывается после того, как клиент ввел значение снимаемой суммы с клавиатуры».
 16. Добавить класс с именем **Устройство выдачи наличных**, для которого выбрать стереотип **boundary**. В секцию документации данного класса следует ввести поясняющий текст: «Устанавливается на банкомате».

17. Для класса *Устройство выдачи наличных* добавить операцию: *выдать наличные()* с квантором видимости **public**. В секцию документации данной операции следует ввести поясняющий текст: «Вызывается после того, как выполнено снятие запрошенной клиентом суммы со счета».
18. Добавить класс с именем *Принтер Банкомата*, для которого выбрать стереотип **boundary**. В секцию документации данного класса следует ввести поясняющий текст: «Устанавливается на банкомате».
19. Для класса *Принтер Банкомата* добавить операцию: *распечатать чек()* с квантором видимости **public**. В секцию документации данной операции следует ввести поясняющий текст: «Вызывается по дополнительному запросу клиента».
20. Добавить направленную ассоциацию от класса *Контроллер Банкомата* к классу *Устройство чтения карточки*. В качестве кратности концов этой ассоциации установить значение 1.
21. Добавить направленную ассоциацию от класса *Контроллер Банкомата* к классу *Принтер Банкомата*. В качестве кратности концов этой ассоциации установить значение 1.
22. Добавить направленную ассоциацию от класса *Контроллер Банкомата* к классу *Клавиатура Банкомата*. В качестве кратности концов этой ассоциации установить значение 1.
23. Добавить направленную ассоциацию от класса *Контроллер Банкомата* к классу *Устройство выдачи наличных*. В качестве кратности концов этой ассоциации установить значение 1.
24. Добавить направленную ассоциацию от класса *Контроллер Банкомата* к классу *Экран Банкомата*. В качестве кратности концов этой ассоциации установить значение 1.
25. Добавить направленную ассоциацию от класса *Контроллер Банкомата* к классу *Контроллер Банка*. В качестве кратности конца этой ассоциации для первого класса установить значение 0..n, а кратности конца ассоциации для второго класса установить значение 1. В качестве стереотипа данной ассоциации выбрать из вложенного списка значение <<communicate>>. Применение данного стереотипа означает, что между этими классами должна существовать физическая взаимосвязь.

Выполнить эти действия предлагается читателям самостоятельно. Построенная в результате указанных действий диаграмма классов будет иметь следующий вид (рис. 6.7).

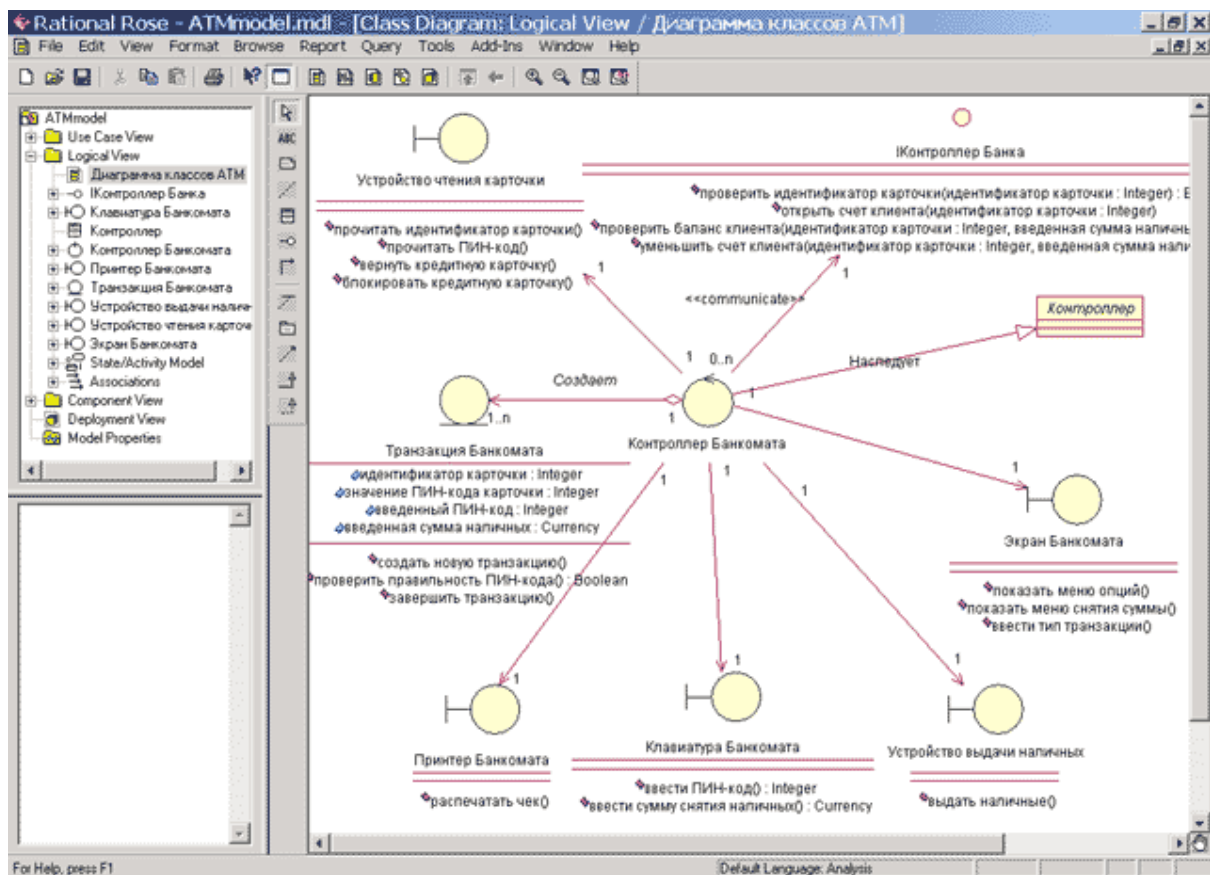


Рис. 6.7. Окончательный вид диаграммы классов для разрабатываемой модели банкомата

Следует заметить, что при изображении диаграммы классов все классы представлены в форме графических стереотипов, при этом выбран способ отображения сигнатуры операций классов. Для более компактного представления диаграммы можно убрать отображение атрибутов, операций или сигнатуры операций отдельных классов с помощью соответствующих операций контекстного меню **Options** (Настройка).