

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ**  
**Государственное образовательное учреждение высшего**  
**профессионального образования**  
**«Казанский национальный исследовательский технический**  
**университет им. А.Н.Туполева»**

**Кафедра автоматизированных систем обработки информации и**  
**управления**

**В.А.СУЗДАЛЬЦЕВ, С.А. ЗАРАЙСКИЙ**

**Б1.В.15 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ИНФОРМАЦИОННЫХ**  
**СИСТЕМ**

**(4308-09 09.03.02 ИСиТ)**

**Методические указания к**  
**лабораторной работе № 4**

**Изучение взаимодействия объектов**  
**(8 часовая работа)**

**Казань – 2016**

### **Цели лабораторной работы.**

1. Изучить теоретическую часть.
2. Повторить пример построения диаграммы классов из лабораторной работы.
3. Выполнить построение диаграммы последовательностей (class diagrams) для своего варианта задания учебного проекта.

### **Теоретические положения.**

#### **Что такое реализация прецедентов**

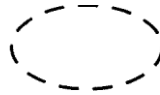
Диаграмма прецедентов (use case diagrams) представляет внешний вид системы. Выполнение прецедентов отображается с помощью потока событий. Сценарии используются для описания того, как реализуются прецеденты, взаимодействуя между группами объектов.

**Сценарий** (scenario) - это элемент прецедента. Он представляет собой одиночный проход по потоку событий для прецедентов. Сценарии помогают выделить **объекты, классы и взаимодействия объектов**, необходимые для исполнения единичного действия, определенного прецедентом. **Сценарии** описывают порядок того, как обязанности, возложенные на прецеденты, распределяются среди объектов и классов в системе. **Сценарии говорят на языке конечных пользователей и экспертов и поэтому являются средством выражения их пожеланий по необходимому поведению системы для разработчиков.**

Каждый прецедент - это сплетение первичных (нормальный поток для прецедента) и **вторичных сценариев** (логика ЧТО-ЕСЛИ в прецеденте). Это значит, что существует множество **сценариев** для системы - **первичные** и **вторичные сценарии** для всех прецедентов. На этапе анализа уже можно сказать, что определение первичного сценария для каждого выбранного прецедента будет достаточным. Когда вы обнаружите, что каждый новый сценарий повторяет большинство шагов из предыдущего, то вы добились цели. **Данная фаза анализа должна завершаться по мере того, как разработчики продумают приблизительно 80% первичных сценариев и выборочно коснутся вторичных.** Если проработать больше сценариев, результаты анализа, вероятно, окажутся хуже; если меньше - не будет достаточного понимания поведения системы, чтобы правильно оценить риски.

По методологии Rational Unified Process реализации прецедентов (**use case realizations**) отражаются в логическом представлении модели. Обратимся к концепции стереотипов, чтобы показать, что прецеденты, созданные в логическом представлении, являются реализациями прецедентов из представления use case. **Другими словами, прецеденты в логическом представлении имеют те же имена, что и в представлении use case, а также стереотип, указывающий на реализацию. В языке UML реализация прецедентов изображается в виде пунктирного овала (см. рис. 1). Логическое представление прецедентов обычно отображается на**

диаграмме прецедентов (или наборе диаграмм), содержащейся в логическом представлении модели.



Название прецедента

Рис. .1. Нотация языка UML для реализации прецедента

Для каждого претендента должен быть создан пакет **Use-Case Realization**.

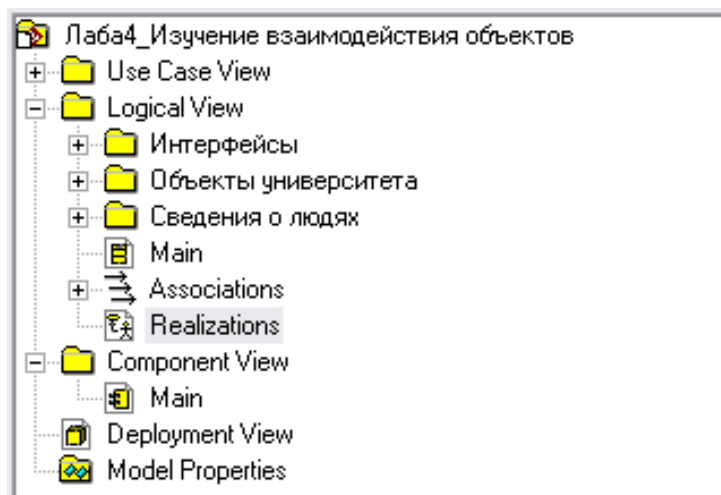


Рис. 2. Окно браузера с диаграммой реализации претендентов (Realizations).

Последовательность создания реализаций прецедентов в программе Rational Rose:

Создание диаграммы прецедентов в логическом представлении модели в программе Rational Rose состоит из следующих шагов:

1. Щелкните правой кнопкой мыши по папке **Logical View** (Логическое представление) в окне браузера.

2. В появившемся контекстно-зависимом меню выберите команду **New => Use Case** (Создать => Прецедент). В раздел логического представления модели будет добавлена новая диаграмма прецедентов с названием **New Diagram**.

3. Введите для новой диаграммы название **Realizations**.

Окно браузера с диаграммой прецедентов **Realizations** показано на рис.

2.

1. Дважды щелкните по диаграмме прецедентов **Realizations** в списке браузера, чтобы открыть диаграмму.

2. Щелкните по кнопке **Use Case** (Прецедент) на панели инструментов.

3. Щелкните по диаграмме прецедентов.

В диаграмму и список браузера будет добавлен новый прецедент.

4. Дважды щелкните по изображению прецедента. На экране появится диалоговое окно **Use Case Specification** (Параметры прецедента).

5. Введите название прецедента (такое же, как у модели) в поле ввода **Name** (Имя). Заметьте, что вы должны указать название в диалоговом окне параметров прецедента или в браузере, чтобы сообщить программе Rational Rose об использовании **другого пространства имен** (namespace). Если вы введете название прецедента непосредственно на диаграмме, программа Rational Rose будет считать, что это тот же прецедент, что и в представлении use case.

6. В открывающемся списке **Stereotype** (Стереотип) выберите стереотип **use- i case realization**.

7. Щелкните по кнопке **OK**, чтобы закрыть диалоговое окно.

Диаграмма прецедентов **Realizations** показана на рис. 3.

Связь между прецедентами в логическом и use case-представлении отражается : путем добавления прецедентов из представления use case на диаграмму **Realizations** и соединения с их реализациями посредством однонаправленной ассоциативной связи с соответствующим стереотипом. (В языке UML используются стереотипные зависимости, но они пока не поддерживаются программой Rational Rose). Для этого необходимо создать новую диаграмму вариантов использования с названием Realization Traceabilities (стандартное название **Use-Case Diagram** для связи между **Use-Case** и **Use-Case Realization**).

На рис. 4 показана связь реализаций с представлением прецедентов на диаграмме функций **Realizations**.

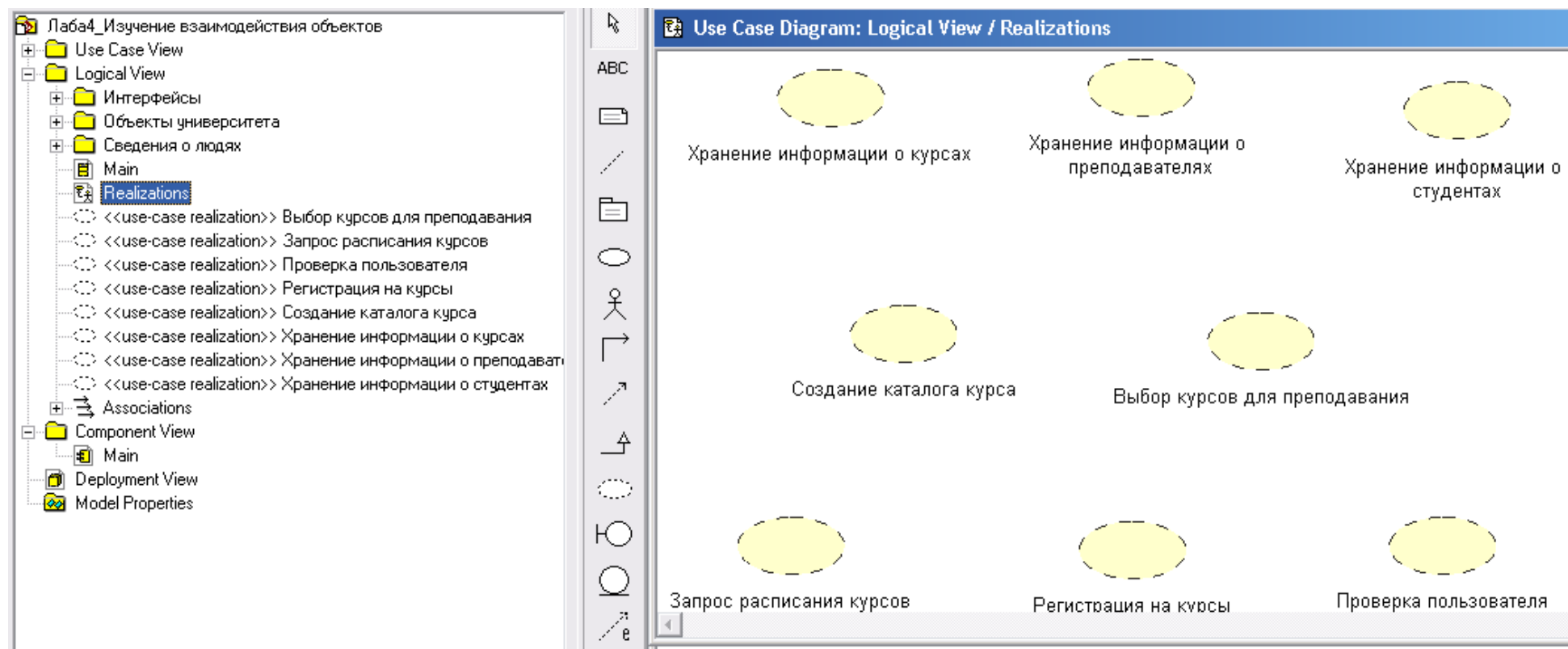


Рис. 3. Диаграмма реализации претендентов.

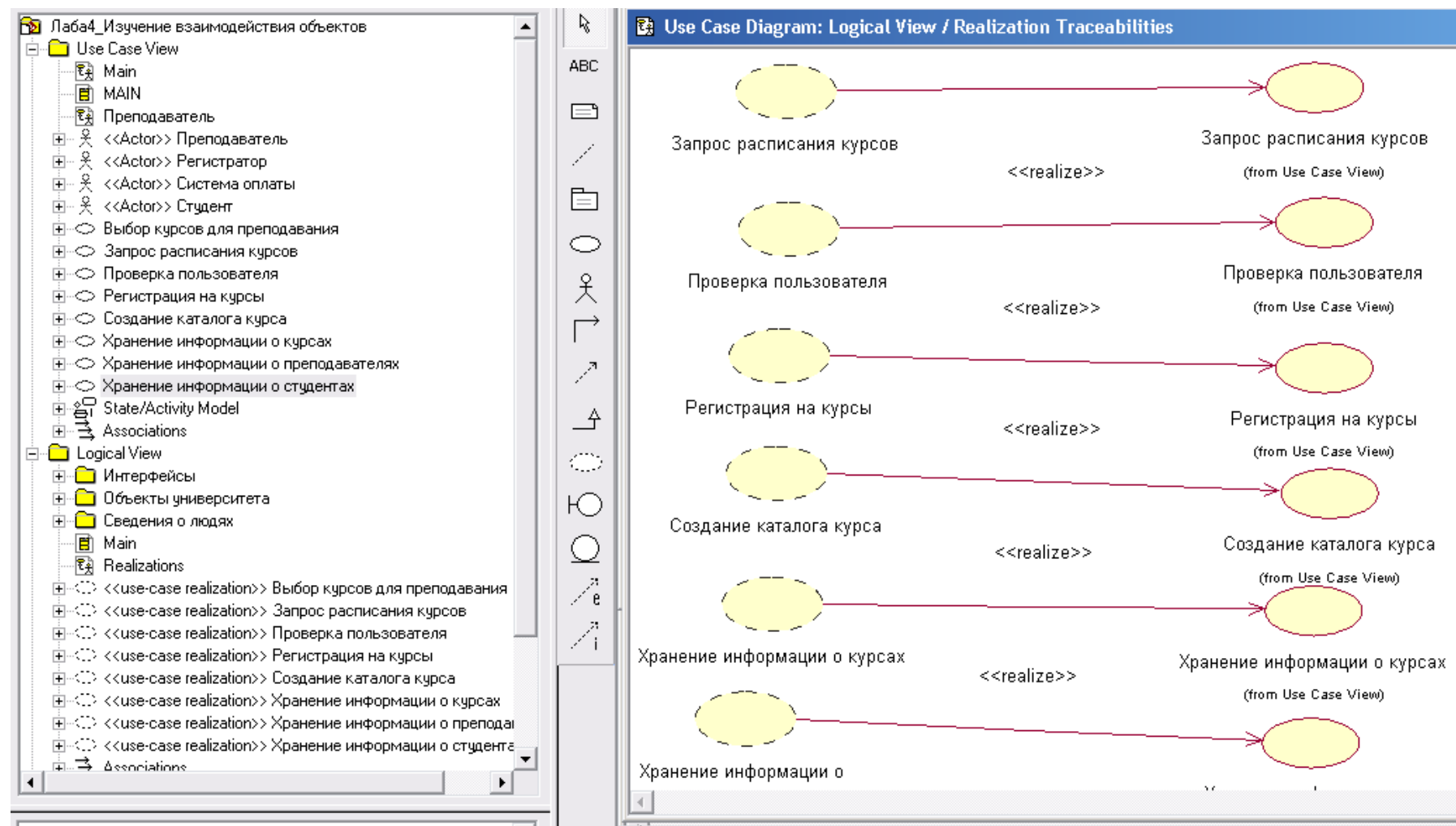


Рис. 4. Связь реализаций с представлением прецедентов.

## Документирование сценариев

Поток событий для прецедента описывается словами, тогда как сценарии отображаются с помощью **диаграмм взаимосвязи** (interaction diagrams). Существует два типа **диаграмм взаимосвязи** - **диаграммы последовательности действий** (sequence diagrams) и **диаграммы взаимодействий** (collaboration diagrams). Каждая диаграмма является графическим представлением сценария

### Диаграммы последовательности действий

**Диаграмма последовательности действий** (sequence diagram) отображает взаимодействие объектов, упорядоченное по времени. На ней показаны **объекты и классы**, используемые в сценарии, и **последовательность сообщений**, которыми обмениваются объекты, для выполнения сценария. Диаграммы последовательности действий обычно соответствуют реализациям прецедентов в логическом представлении системы.

В языке UML объект на **диаграмме последовательности действий** выглядит как прямоугольник, содержащий подчеркнутое название объекта. Название может состоять только из имени объекта, из имени объекта и его класса или только имени класса (анонимный объект). Эти три вида наименований объекта показаны на рис. 5.

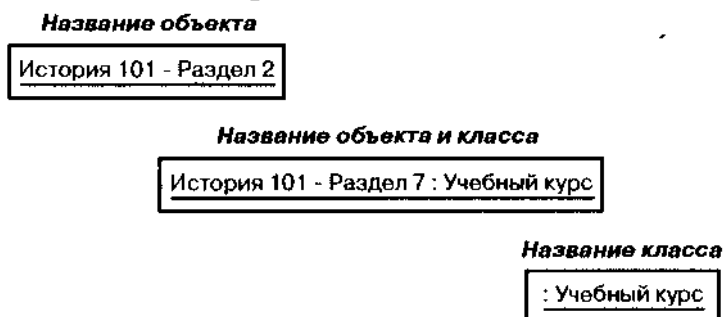


Рис. 5. Наименования объектов на диаграмме последовательности действий

Названия объектов могут быть конкретными (например, Алгебра 101, раздел 1) или общими (например, учебный курс). Часто анонимные объекты используются для представления любого объекта данного класса.

Каждый объект также имеет свою временную линию (timeline), изображаемую пунктиром под объектом. Сообщения, передаваемые между объектами, указываются стрелками, направленными от клиента (отправителя сообщения) к поставщику (получателю сообщения).

На рис. 6 представлена нотация языка UML для объектов и сообщений на **диаграмме последовательности действий**.

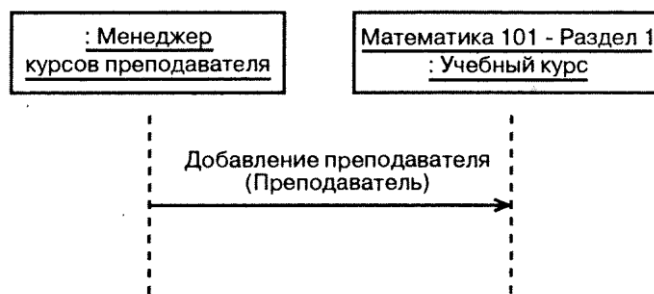


Рис. 6. Нотация языка UML для объектов и сообщений на диаграмме последовательности действий

Для создания диаграммы последовательности действий в программе Rational Rose:

1. Щелкните правой кнопкой мыши по папке **Logical View** (Логическое представление) в окне браузера.
  2. В появившемся контекстно-зависимом меню выберите команду **New = Sequence Diagram** (Создать => **Диаграмма последовательности действий**). В список браузера будет добавлена новая диаграмма.
  3. Введите ее имя (например, «Создание учебного предмета»).
- Окно браузера с диаграммой последовательности действий показано на рис. 7.

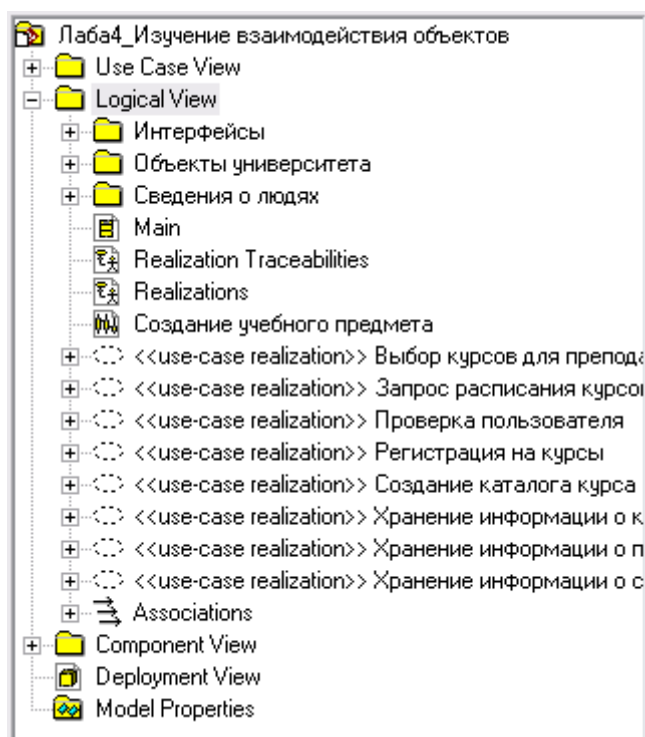


Рис. 7. Окно браузера с диаграммой последовательности действий (Создание учебного предмета).

Чтобы создать объекты и сообщения на диаграмме последовательности действий в программе Rational Rose:

1. Дважды щелкните по диаграмме последовательности действий в списке браузера, чтобы открыть диаграмму.



2. Выберите из списка актера, щелкнув по нему мышью.
3. Перетащите актера на диаграмму последовательности действий.
4. Щелкните по кнопке **Object** (Объект) на панели инструментов.
5. Щелкните по диаграмме последовательности действий, чтобы добавить новый объект.
6. Введите имя объекта.
7. Повторите предыдущие шаги для каждого объекта и актера в сценарии.
8. Щелкните по кнопке **Object Message** (Сообщение) на панели инструментов.
9. Щелкните по актеру или объекту-отправителю сообщения и проведите стрелку сообщения к актеру или объекту-получателю.
10. Введите название сообщения.
11. Повторите шаги с седьмого по девятый для каждого сообщения в сценарии.
12. Диаграмма последовательности действий для сценария создание учебного предмета (Create a Course) изображена на рис. 8.

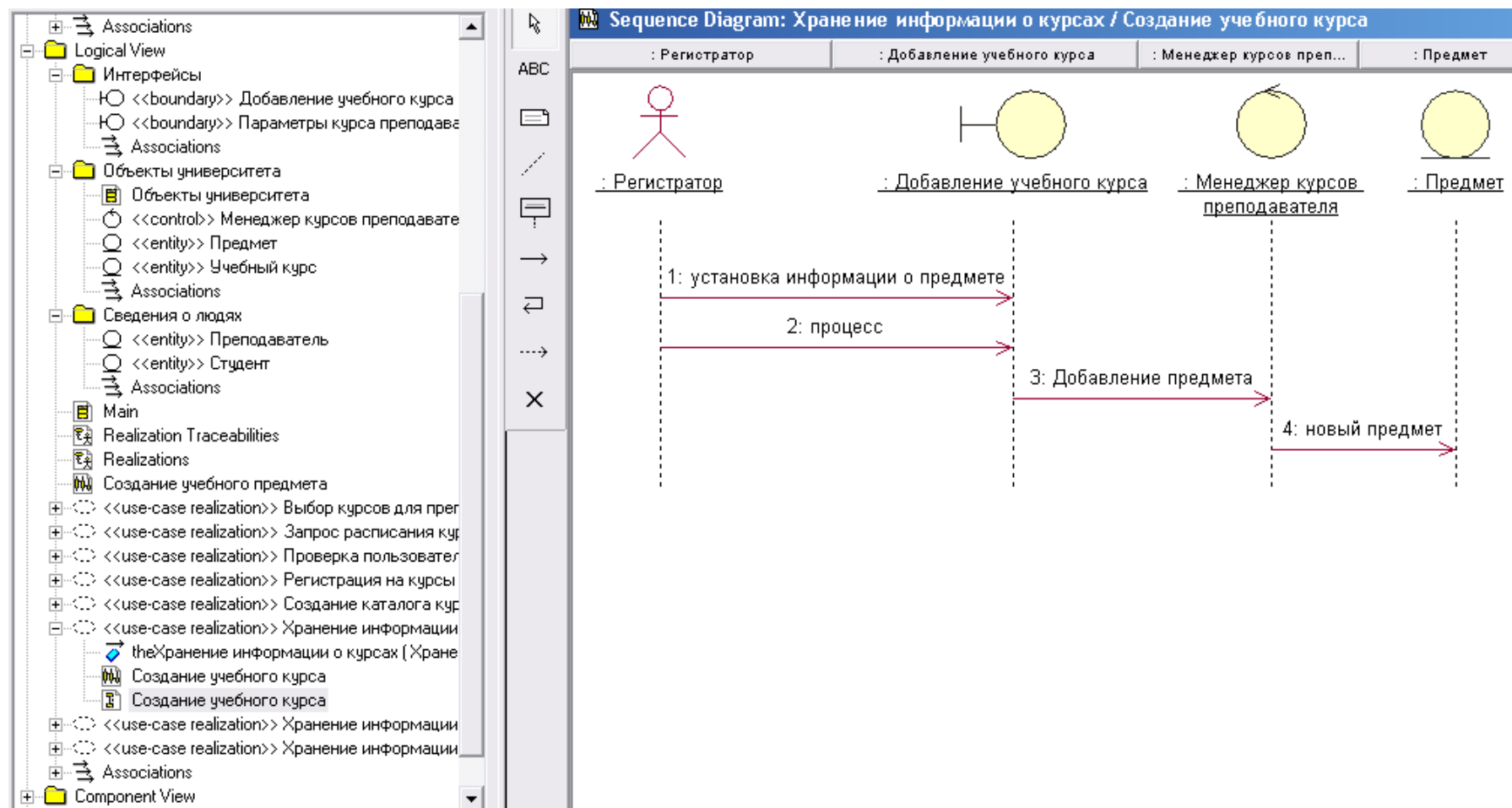


Рис. 8. Диаграмма последовательности действий для сценария создания учебного курса

## Диаграммы последовательности действий и граничные классы

Граничные классы добавляются на диаграмму последовательности действий для того, чтобы показать взаимодействие с пользователем или другой системой. На стадии анализа назначение граничных классов на диаграмме заключается в описании требований к интерфейсу, но не в описании реализации интерфейса.

Реальные сообщения, поступающие от **актера граничному классу**, и информация об их последовательности зависят от структуры приложения и определяются на стадии проектирования. Они могут изменяться, по мере того как в систему добавляется информация о способах реализации.

### Сложность и диаграммы последовательности действий

Вопрос: «Насколько сложной может быть диаграмма последовательности действий?»

Ответ: «Сохраняйте ее простой».

Прелесть этой диаграммы в ее простоте - можно легко понять и увидеть объекты, взаимодействия объектов, сообщения между ними и функциональность, задаваемую сценариями.

Следующий вопрос выглядит так: «Что делать с условной логикой?» (с логикой ЕСЛИ-ТО-ИНАЧЕ, которая существует в реальном мире). Ответ на него также субъективен. Если логика проста и требует небольшого количества сообщений, то обычно добавляется ее к одной диаграмме и использую примечания и условия для указания выбора, который нужно сделать. С другой стороны, **если логика ЕСЛИ- ТО-ИНАЧЕ требует сложных сообщений, то обычно рисуют отдельные диаграммы: ; одну для случая ЕСЛИ, одну для ТО и одну для ИНАЧЕ.**

Это делается для сохранения простоты диаграмм. Если нужно, их можно связать друг с другом. Это позволит пользователям перемещаться по набору диаграмм.

Для связывания диаграмм в программе Rational Rose:

1. Щелкните по кнопке Note (Сноска) на панели инструментов.
2. Щелкните по диаграмме, чтобы поместить на нее сноску.
3. Выберите в списке браузера диаграмму, которую нужно связать с текущей, и перетащите ее на сноску.
4. Для перехода на связанную диаграмму необходимо дважды щелкнуть по сноске.

### Диаграммы взаимодействий

**Диаграмма взаимодействий** (collaboration diagram) - это альтернативный способ отображения сценариев. Такой тип диаграммы показывает взаимодействие объектов, организованное вокруг них, и их связи

друг с другом. Диаграмма взаимодействий содержит:

- объекты, изображаемые в виде прямоугольников;
- связи между объектами, изображаемые в виде линий;
- сообщения в виде текста и стрелки, направленной от клиента к поставщику.

Нотация языка UML для объектов, связей и сообщений на диаграмме взаимодействий показана на рис. 10.

Последовательность создания диаграмм взаимодействий из диаграмм последовательности действий в программе Rational Rose:

1. Дважды щелкните по диаграмме последовательности действий в списке браузера, чтобы открыть диаграмму.
2. Выберите команду меню **Browse => Create collaboration diagram** (Просмотр => Создать диаграмму взаимодействий) или нажмите клавишу **F5**.
3. Расположите объекты и сообщения на диаграмме нужным образом.

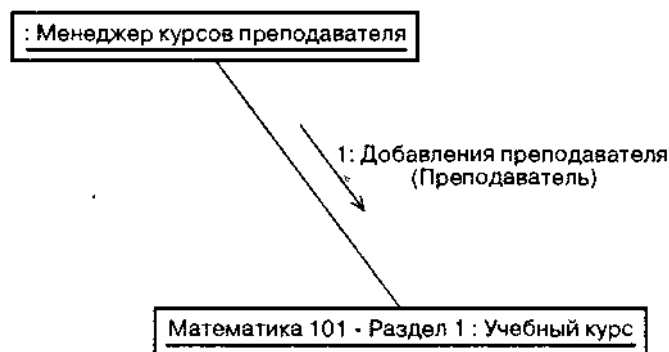


Рис. 10. Нотация языка UML для объектов, связей и сообщений на диаграмме взаимодействий

Диаграмма взаимодействий показана на рис. 11.

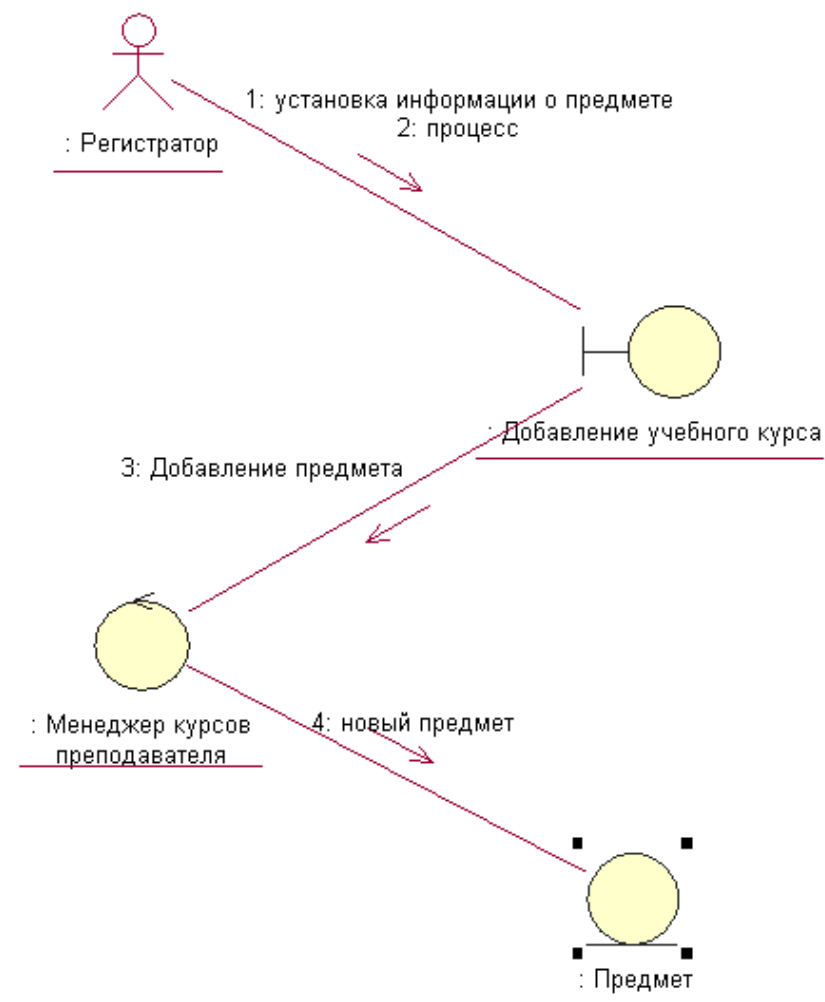
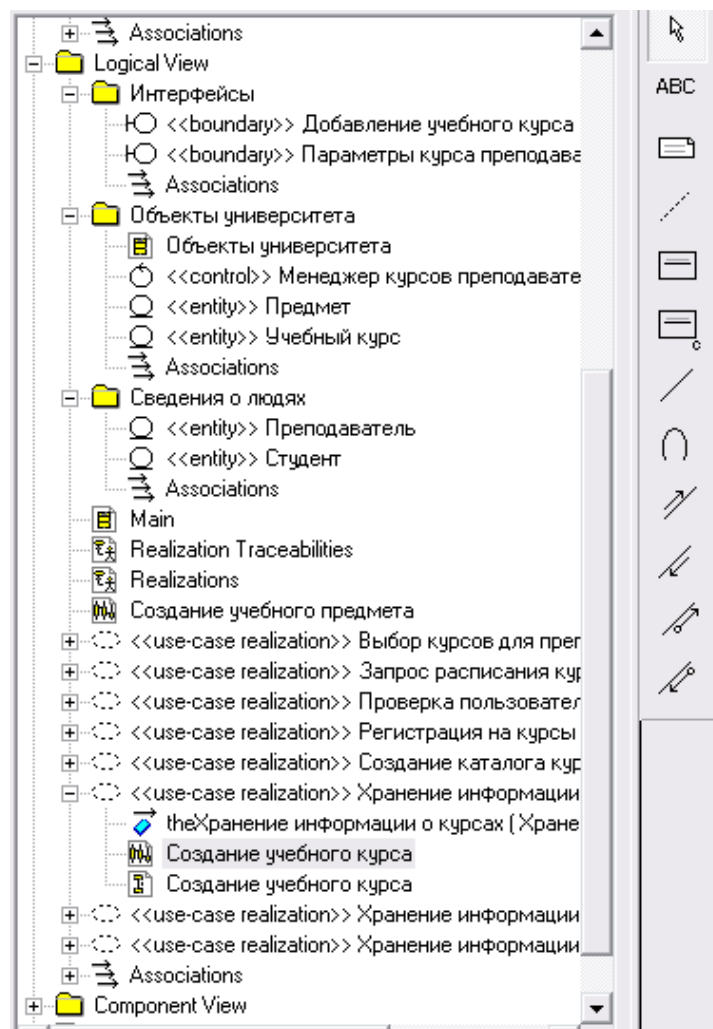


Рис. 11. Диаграмма взаимодействий

Можно сначала создать диаграмму взаимодействий. В этом случае диаграмма последовательности действий может быть получена из нее. Для этого необходимо выбрать команду меню **Browse => Create Sequence Diagram** (Просмотр => Создать диаграмму последовательности действий) или нажать клавишу **F5**.

### **Зачем нужны две разные диаграммы**

Диаграмма последовательности действий используется для просмотра сценария во временном порядке: что происходит сначала, что происходит затем. Заказчики легко могут читать и понимать такие диаграммы. Поэтому они очень полезны на **стадии анализа**. Диаграмма взаимодействий представляет общую картину сценария, так как взаимодействия на ней организованы между связанными друг с другом объектами. Такой тип диаграмм чаще используется на этапе проектирования, когда планируется реализация отношений.

### **Диаграмма последовательности действий для системы регистрации курсов**

Продолжим анализ сценария добавление учебного курса (Add a Course Offering). Диаграмма показана на рис. 12.

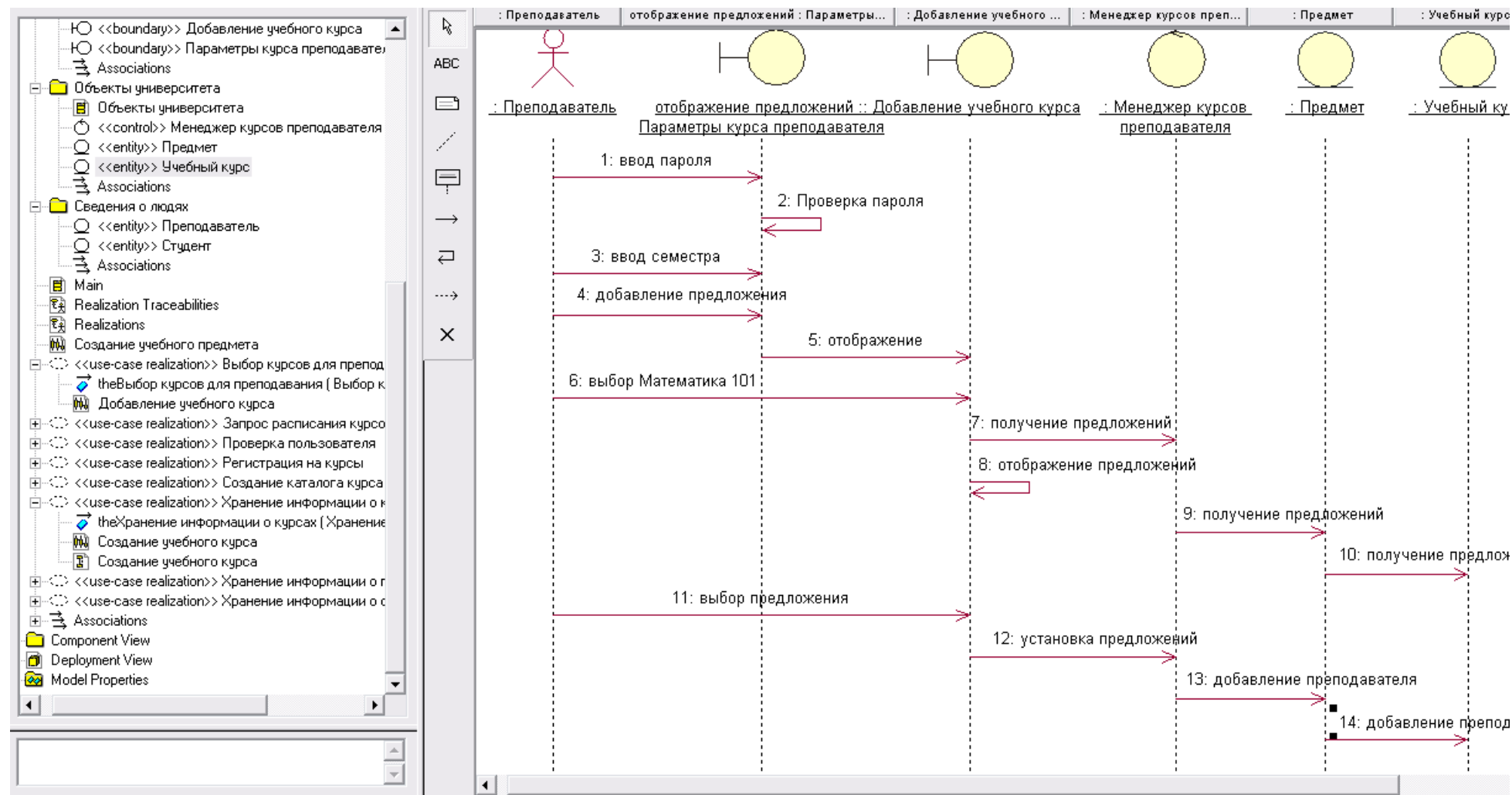


Рис. 12. Диаграмма последовательности действий для сценария добавление учебного курса

## **ЗАКЛЮЧЕНИЕ**

Диаграмма прецедентов представляет внешний вид системы. Выполнение прецедентов отображается с помощью потока событий. Сценарии используются для описания того, как прецеденты реализуются в виде взаимодействия между группами объектов. Сценарий - это экземпляр прецедента. Он представляет собой одиночный проход по потоку событий для прецедента. Таким образом, каждый прецедент - это сплетение сценариев. Они помогают выделить объекты, классы и взаимодействия объектов, необходимые для исполнения единичного действия, определенного прецедентом.

Поток событий для прецедентов обычно описывается словами, тогда как сценарии - диаграммами взаимосвязи. Существует два типа диаграмм взаимосвязи - диаграммы последовательности действий (sequence diagrams) и диаграммы взаимодействий (collaboration diagrams). Каждая диаграмма - это графическое представление сценария.

Диаграмма последовательности действий отображает взаимодействие объектов, упорядоченное по времени. Диаграмма взаимодействий - это альтернативный способ отображения сценариев. Этот тип диаграммы показывает взаимодействие объектов, организованное вокруг самих объектов, и их связи друг с другом.

## **2. Порядок выполнения лабораторной работы**

1. Изучить методические материалы по лабораторной работе №4.
2. Повторить все диаграммы лабораторной работы.
3. Составить диаграмму последовательности действий для своего варианта задания.
4. Оформить лабораторную работу. Отчёт составляется в печатной (на бумаге) форме, каждому студенту. Отчет включает описание всей проделанной работы (интерпретацию постановки задачи, глоссарий проекта, дополнительные спецификации, описание элементов диаграммы вариантов использования) и скриншот полученной диаграммы.
5. Защитить лабораторную работу. Защита работы сопровождается необходимыми пояснениями и подробными и точными ответами на контрольные вопросы из нижеприведённого списка.



### **Контрольные вопросы.**

1. Что такое объект в UML?
2. Что такое состояние, поведение и индивидуальность объекта
3. Что такое класс в UML?
4. Что такое «хороший класс»?
5. Стереотипы и классов?
6. Что значит документировать класс? Примеры.
7. Пакеты. Назначение.
8. Диаграммы последовательности. Назначение.

### **Литература:**

1. Кватрани Т. Rational Rose 2000 и UML. Визуальное моделирование: Пер. с англ. - М.: ДМК Пресс, 2001. - 176 с.: ил. (Серия «Объектно-ориентированные технологии в программировании»).
2. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. Учебник. М.: Финансы и статистика, 2003, 352 с.
3. Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем. М.: Финансы и статистика, 2002, 192 с.

