Моделирование реализации вариантов использования

Диаграммы взаимодействия (collaboration diagrams)

Лекция 8

Назначение диаграммы кооперации

Характерной особенностью диаграммы кооперации является визуализация последовательности сообщений, описывающих процесс динамического взаимодействия объектов.

На диаграмме кооперации поведение системы описывается на уровне отдельных объектов, которые обмениваются между собой сообщениями, чтобы достичь нужной цели или реализовать некоторый вариант использования.

Понятие кооперации (collaboration)

Цель кооперации – специфицировать особенности реализации отдельных вариантов использования или наиболее значимых операций в системе.

Кооперация может быть представлена на двух уровнях:
- на уровне спецификации
- на уровне примеров

Диаграмма кооперации уровня спецификации показывает роли, которые играют участвующие во взаимодействии элементы. Элементами кооперации на этом уровне являются множества объектов (классы) и множества связей (ассоциации), которые обозначают отдельные роли классификаторов и роли ассоциаций между участниками кооперации.

Диаграмма кооперации уровня примеров (instances) визуализирует объекты (экземпляры классов), связи (экземпляры ассоциаций) и сообщения. При этом связи дополняются стрелками сообщений.

Объект (object)

Объект является отдельным экземпляром класса, который создается на этапе реализации модели (выполнения программы).

Он может иметь свое собственное имя и конкретные значения атрибутов.

Полное имя объекта:
<собственное имя объекта>'/'<Имя роли классификатора>:< Имя классификатора >

Варианты возможных записей имен объектов

o : C — объект с собственным именем o, образуемый на основе класса C;

: C — анонимный объект, образуемый на основе класса C;

o : (или просто o) — объект-сирота с собственным именем o;

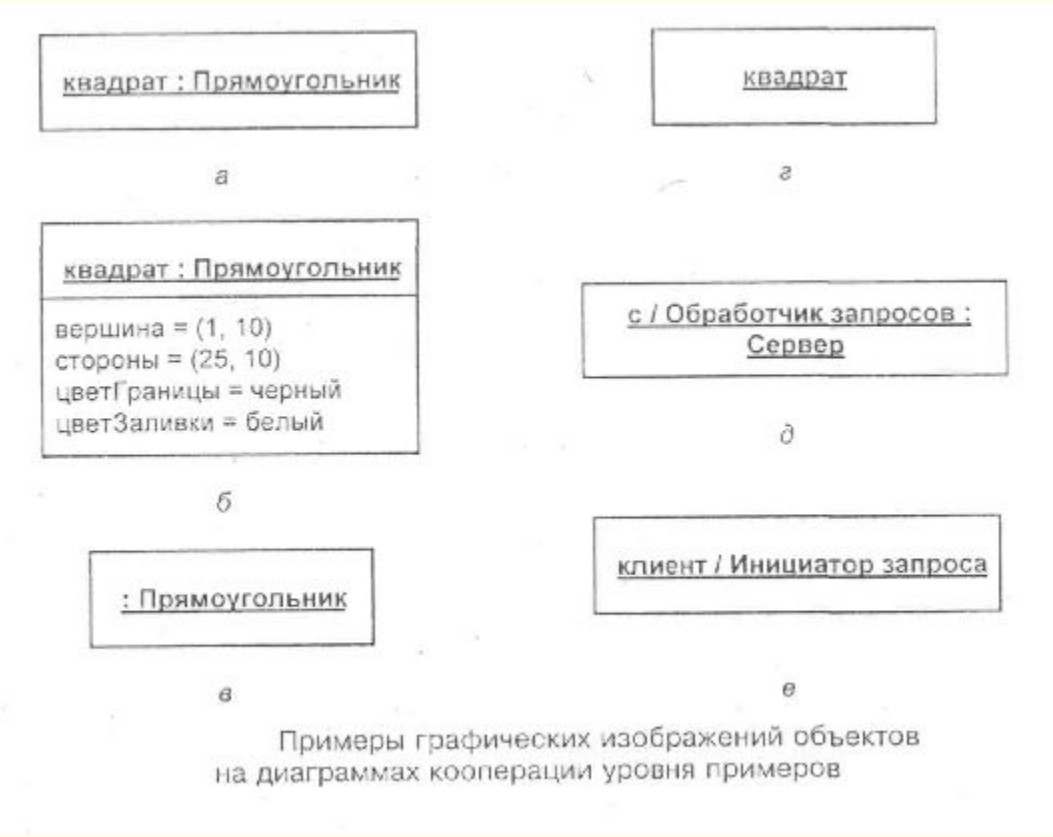
o / R : C — объект с собственным именем o, образуемый на основе класса C и играющий роль R;

/ R : C — анонимный объект, образуемый на основе класса C и играющий роль R;

o / R — объект-сирота с собственным именем o, играющий роль R;

/ R — анонимный объект и одновременно объект-сирота, играющий роль R.

Примеры изображения и именования объектов диаграммы кооперации

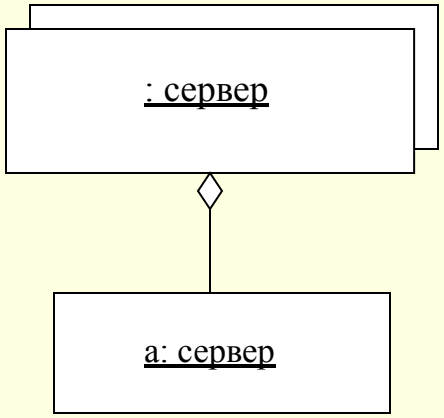


Примеры графических изображений объектов на диаграммах кооперации уровня примеров

Мультиобъект (multiobject)

Мультиобъект:

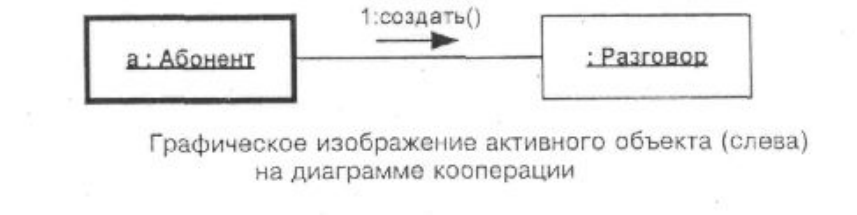
- множество объектов, которые могут быть образованы на основе одного класса;
- используется для того, чтобы показать операции и сигналы, которые адресованы всему множеству объектов, а не только отдельному объекту;
- изображается двумя прямоугольниками (один выступает из-за другого), при этом стрелка взаимосвязи относится ко всему множеству объектов.



Активный объект

Пассивный объект оперирует только данными и не может инициировать деятельность по управлению другими объектами.

Активный объект имеет свой собственный поток управления, может инициировать деятельность по управлению другими объектами.



Графическое изображение активного объекта (слева) на диаграмме кооперации

Фрагмент диаграммы кооперации для вызова функции печати из текстового редактора

Составной объект

Составной объект (composite object) или объект-композит предназначен для представления объекта, имеющего собственную структуру и внутренние потоки управления.
Составной объект является экземпляром класса-композита, который связан отношением композиции со своими частями.

Графическое изображение составного объекта на диаграмме кооперации

Понятие связи и ее стереотипа

Виды стереотипов:

«association» - связь-ассоциация, предполагается по умолчанию, поэтому этот стереотип может не указываться;

«parameter» - параметр операции или метода. Соответствующий объект может быть только параметром некоторой операции;

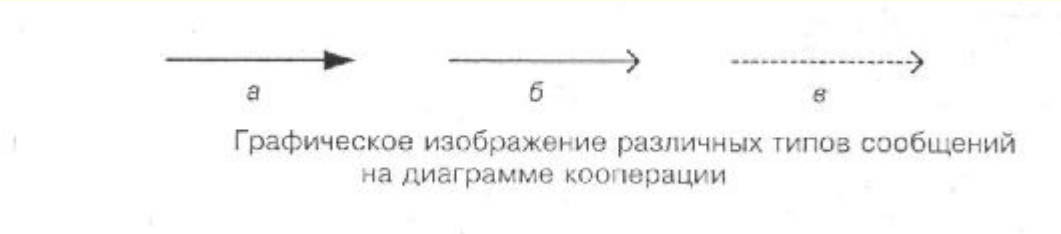
«local» - локальная переменная. Ее область видимости ограничена только соседним объектом;

«global» - глобальная переменная. Ее область видимости распространяется на все объекты диаграммы;

«self» - рефлексивная связь объекта с самим собой, которая допускает передачу объектом сообщения самому себе. На диаграмме изображается петлей.

Сообщение

Сообщение специфицирует коммуникацию между двумя объектами, один из которых передает другому некоторую информацию.
При этом первый объект предполагает, что после получения сообщения вторым объектом последует выполнение некоторого действия.



Графическое изображение различных типов сообщений на диаграмме кооперации

Пример построения диаграммы кооперации системы управления банкоматом

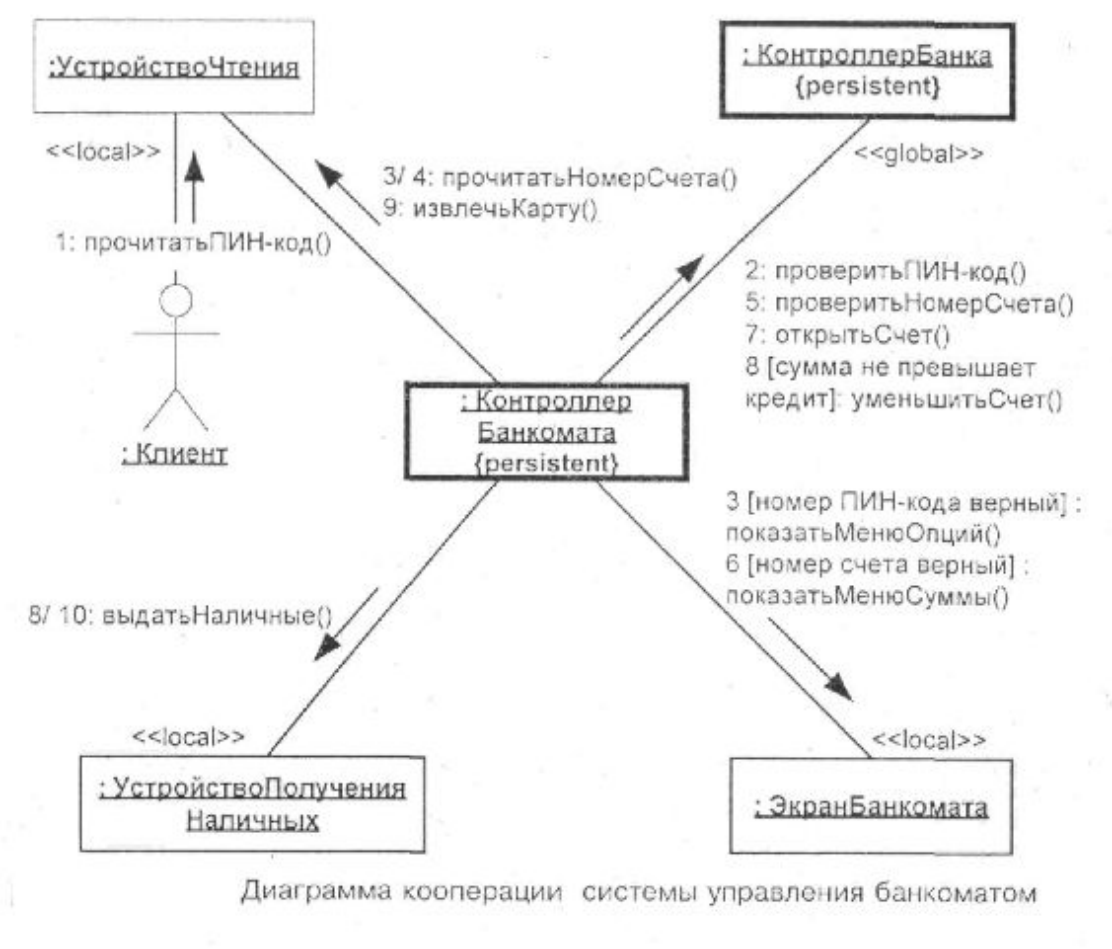
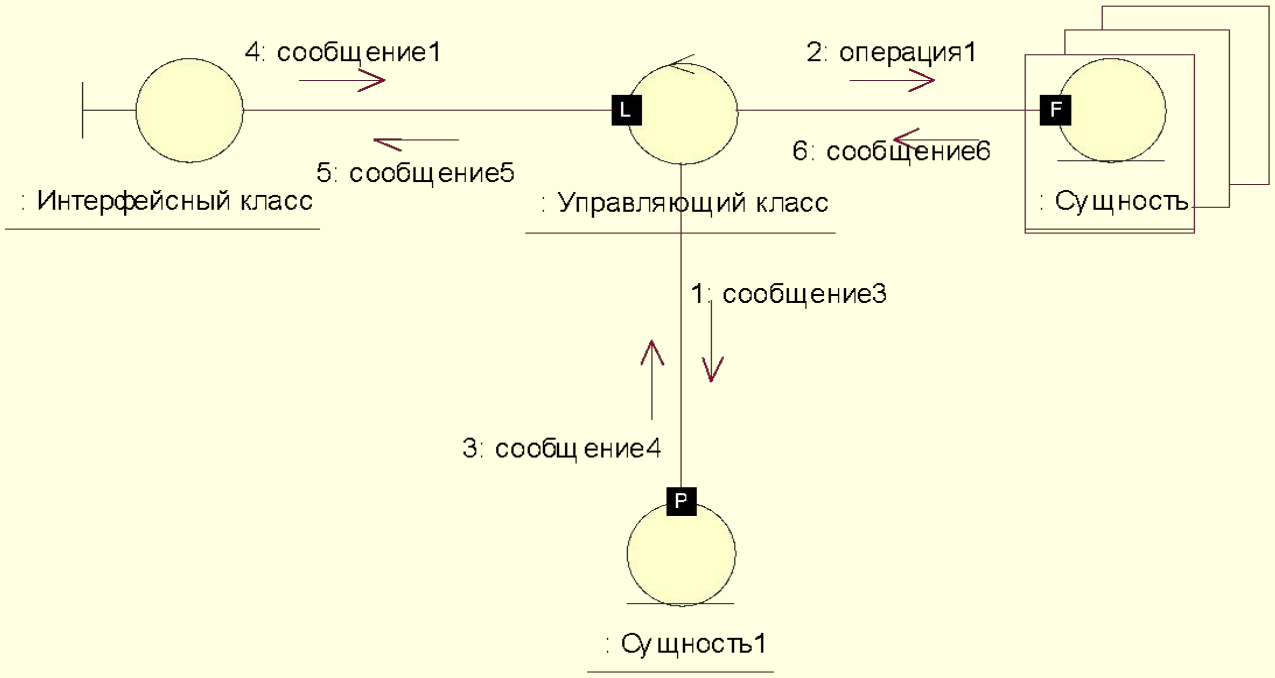


Диаграмма кооперации системы управления банкоматом

Пример диаграммы сотрудничества

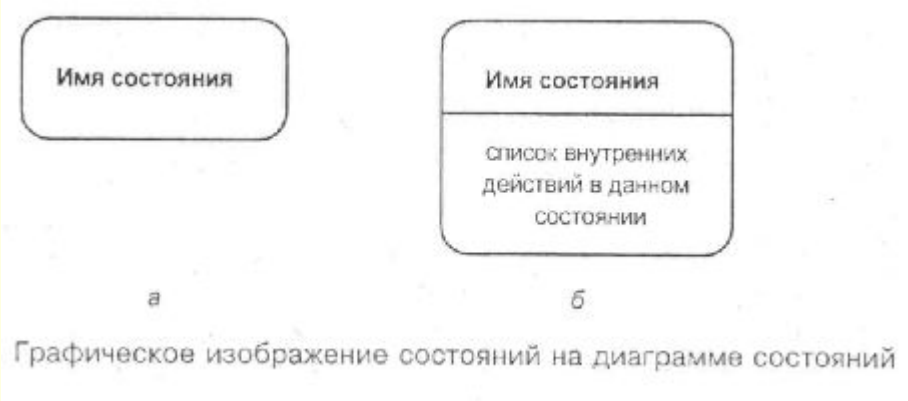


Диаграммы состояний (statechart diagram)

Диаграмма состояний описывает процесс изменения состояний системы или ее подсистемы при реализации всех вариантов использования.
При этом изменение состояний отдельных элементов системы может быть вызвано внешними воздействиями со стороны других элементов или извне системы.

Т.о. для описания реакции системы на подобные внешние воздействия и используются диаграммы состояний.

Состояние и имя состояния



Графическое изображение состояний на диаграмме состояний

Имя состояния представляет собой строку текста, которая раскрывает содержательный смысл данного состояния.

Имя должно представлять собой законченное предложение и всегда записываться с заглавной буквы.

Список внутренних действий

Каждое из действий записывается в виде отдельной строки и имеет следующий формат:

<метка действия '/' выражение действия>

- **entry** – указывает на то, что следующее за ней выражение действия должно быть выполнено в момент входа в данной состояние (входное действие)
- **exit** - указывает на то, что следующее за ней выражение действия должно быть выполнено в момент выхода из данного состояния (выходное действие)
- **do** – специфицирует некоторую деятельность (do activity), которая выполняется в течение всего времени, пока объект находится в данном состоянии, или до тех пор пока не будет выполнено условие ее окончания, специфицированное в соответствующей операции класса или вычислительной процедуре. В последнем случае при завершении деятельности генерируется соответствующее событие.
- **include** – используется для обращения к конечному подавтомату, при этом следующее за ней выражение действия содержит имя этого подавтомата.

Начальное и конечное состояния

Начальное состояние (initial state) – частный случай состояния, которое не содержит никаких внутренних действий и поэтому относится к категории псевдосостояния. В этом состоянии находится объект по умолчанию в начальный момент.

Графическое изображение начального и конечного состояний на диаграмме состояний

Конечное (финальное) состояние (final state) представляет собой частный случай состояния, которое также не содержит никаких внутренних действий и поэтому также является псевдосостоянием. В этом состоянии должен находиться моделируемый объект или система по умолчанию после завершения работы конечного автомата.

Переход

Простой переход (simple transition) представляет собой отношение между двумя последовательными состояниями, которое указывает на факт смены одного состояния другим.

Переход осуществляется при наступлении некоторого события: окончания выполнения деятельности (do activity), получении объектом сообщения или приемом сигнала.

Срабатывание перехода может зависеть не только от наступления некоторого события, но и от выполнения определенного условия, называемого сторожевым условием.

Каждый переход может быть помечен строкой текста, которая имеет следующий общий формат:

<сигнатура события>'['< сторожевое условие>']'/< выражение действия>

Типы переходов

- Переход называется ***триггерным***, если его специфицирует некоторое событие-триггер. В этом случае рядом со стрелкой триггерного перехода обязательно указывается имя события в форме строки текста; в качестве имен триггерных переходов задают имена операций, вызываемых у тех или иных объектов системы.
- Переход называется ***нетриггерным***, если он происходит по завершении выполнения действий (деятельности) в исходном состоянии.

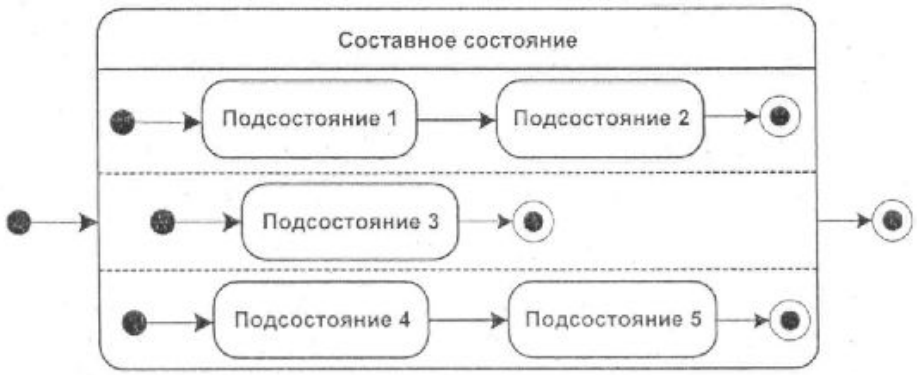
Сторожевое условие и выражение действия

- **Сторожевое условие** (guard condition), если оно есть, всегда записывается в прямых скобках и представляет собой некоторое булевское выражение.
- **Выражение действия** (action expression) выполняется только в том случае, когда переход срабатывает. В общем случае, выражение действия может содержать целый список отдельных действий, разделенных символом ";".

Фрагмент диаграммы состояний
для моделирования почтовой программы-клиента

Составное состояние и подсостояние

- **Составное состояние** (composite state) — сложное состояние, которое состоит из других вложенных в него состояний. Последние выступают по отношению к первому как *подсостояния* (substate).
- **Последовательные подсостояния** (sequential substates) используются для моделирования такого поведения объекта, во время которого в каждый момент объект может находиться в одном и только одном подсостояний.
- **Параллельные подсостояния** (concurrent substates) позволяют специфицировать два и более конечных подавтомата, которые могут выполняться параллельно внутри составного события. Каждый из конечных подавтоматов занимает некоторую область (регион) внутри составного состояния, которая отделяется от остальных горизонтальной пунктирной линией.



Графическое изображение составного состояния с вложенными параллельными подсостояниями

Statechart Diagram - пример