

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Пермская государственная сельскохозяйственная академия
имени академика Д.Н. Прянишникова»

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ
направление 230700 «Прикладная информатика»

ЛАБОРАТОРНОЕ ЗАНЯТИЕ № 5

Тема: **Модель прецедентов: описание системных операций**

Учебные вопросы:

1. Модель прецедентов: диаграммы последовательностей.
2. Модель прецедентов: детализация с помощью описания операций.

Вопрос 1. Модель прецедентов: диаграммы последовательностей

Основное внимание будет уделено анализу прецедента и моделированию предметной области. До начала проектирования целесообразно выполнить дальнейшее исследование предметной области, в частности уточнить входные и выходные события, связанные с данной системой, которые можно проиллюстрировать на диаграмме последовательностей в системе обозначений UML.

Прежде чем приступать к проектированию логики работы программного приложения, необходимо исследовать и определить ее поведение как "черного ящика".

Поведение системы (system behavior) представляет собой описание того, *какие* действия выполняет система, без определения механизма их реализации. Одной из частей такого описания является диаграмма последовательностей. К остальным частям относятся прецеденты и описания системных операций.

Диаграммы последовательностей системы

Прецеденты определяют, как исполнители взаимодействуют с программной системой. В процессе этого взаимодействия исполнителем генерируются события, передаваемые системе, которые представляют собой запросы на выполнение некоторой операции.

Например, кассир, введя идентификатор товара, тем самым предписывает, чтобы система POS записала данные о приобретении товара. Это событие инициирует в системе выполнение некоторой операции.

Следует отделить и проиллюстрировать операции системы, выполнение которых запрашивает внешний исполнитель, поскольку они важны для понимания поведения системы. В качестве системы обозначений в состав языка UML входят **диаграммы последовательностей** (sequence diagram). С их помощью можно проиллюстрировать взаимодействие исполнителя с системой и операции, выполнение которых при этом инициируется.

Диаграмма последовательностей системы (system sequence diagram) – это схема, которая для определенного сценария прецедента¹ показывает генерируемые внешними исполнителями события, их порядок, а также события, генерируемые внутри самой системы. При этом все системы рассматриваются как "черный ящик". Назначение данной диаграммы – отображение событий, передаваемых исполнителями системе через ее границы.

Диаграмму последовательностей нужно создать для основного успешного сценария прецедента, а при необходимости и для наиболее существенных и сложных альтернативных сценариев.

В контексте языка UML нет понятия "диаграмма последовательностей системы", есть просто "диаграмма последовательностей". Это уточнение автор использовал для того, чтобы сделать акцент на рассмотрении системы в виде "черного ящика". В дальнейшем диаграммы последовательностей будут рассмотрены в другом контексте – для иллюстрации взаимодействия разрабатываемых программных объектов.

Пример диаграммы последовательностей

На диаграмме последовательностей для определенного хода событий, описанного в прецеденте, отображаются внешние исполнители, которые взаимодействуют непосредственно с системой, сама система (как "черный ящик"), а также системные события, инициируемые исполнителями (рис.1.1). При этом порядок событий должен соответствовать их последовательности в описании прецедента. Системные события могут включать различные параметры.

Пример, приведенный на рис. 1.1, соответствует основному успешному сценарию прецедента **Оформление продажи**. Он иллюстрирует, что для POS-системы кассир генерирует системные события **makeNewSale**, **enterItem**, **endSale** и **makePayment**.

¹ Сценарий прецедента — это его частный случай или реальный путь его реализации.

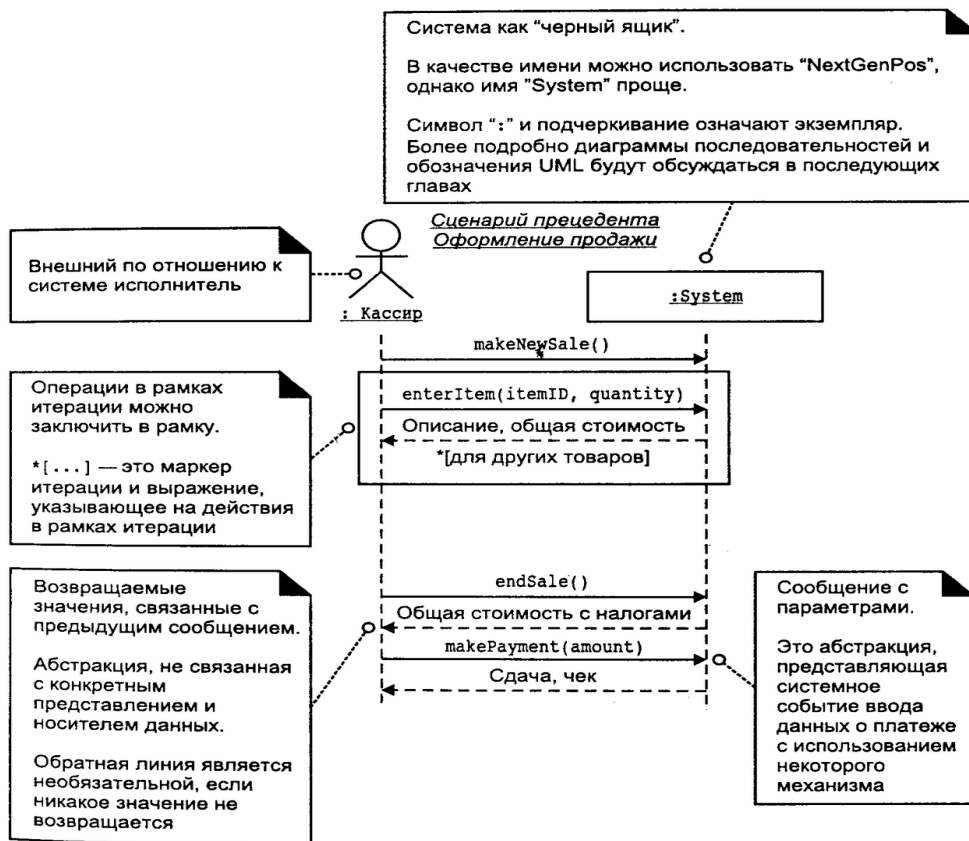


Рисунок 1.1 – Диаграмма последовательностей для сценария прецедента Оформление продажи

Системные события и прецеденты

На диаграмме последовательностей отображаются системные события сценария некоторого прецедента. Поэтому сама диаграмма строится на основе описания прецедента (рис. 1.2).

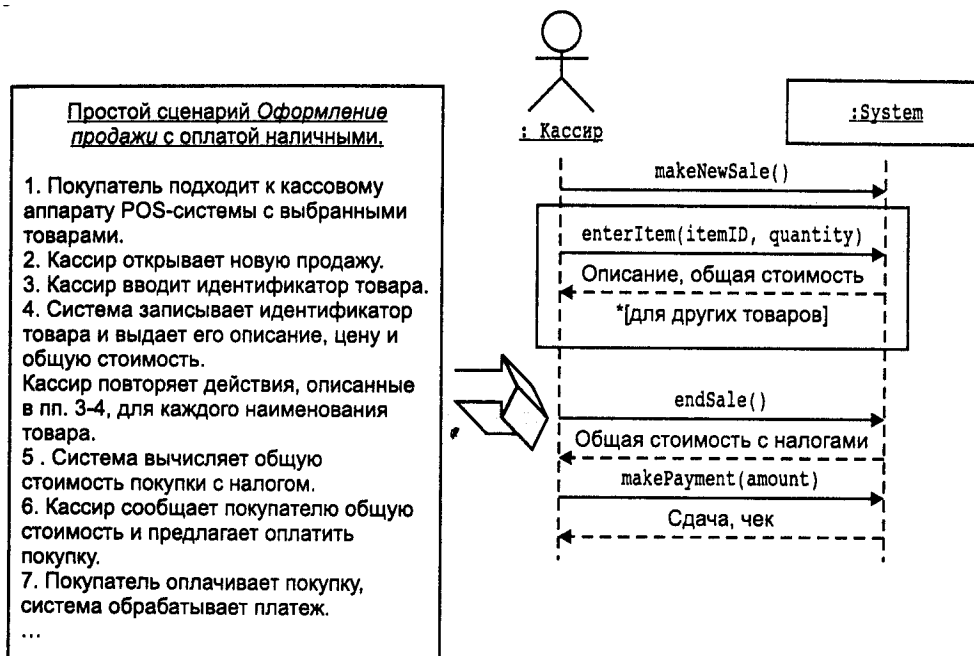


Рисунок 1.2 – Диаграммы последовательностей строятся на основе описания прецедентов

Системные события и границы системы

Как уже упоминалось в предыдущей лабораторной работе, при идентификации системных событий необходимо четко определить, где проходят границы системы. При разработке программной системы ее границы обычно выбираются в соответствии с ее программной

частью (а возможно, и аппаратной). В этом контексте системным является внешнее событие, оказывающее влияние непосредственно на программное обеспечение (рис. 1.3).

Выполним идентификацию системных событий для прецедента Оформление продажи. Сначала необходимо определить исполнителей, которые взаимодействуют с программной системой непосредственно. Покупатель взаимодействует с кассиром, однако при этом его связь с программным обеспечением POS-системы отсутствует. Этот процесс осуществляется только кассиром. Следовательно, генератором системных событий является лишь кассир.

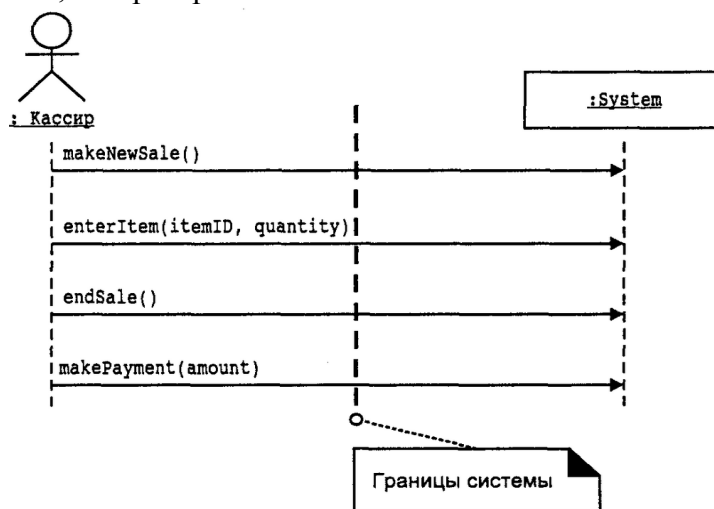


Рисунок 1.3 – Определение границ системы

Имена системных событий и операций

Системным событиям (и связанным с ними системным операциям) имена нужно присваивать осмысленно, а не в терминах входных физических носителей данных или управляющих элементов пользовательского интерфейса.

Имя системного события лучше всего начинать с глагола: **add**. . . (добавить), **enter**... (ввести), **end**... (завершить), **make**... (выполнить) (рис. 1.4). Это повышает читабельность имен и, кроме того, дополнительно подчеркивает направление передачи событий.

Таким образом, имя **enterItem** гораздо лучше, чем имя **scan** (в смысле считывания лазерным сканером), поскольку в нем отражается смысл операции. К тому же такое имя является абстрактным, не зависящим от выбора управляющих элементов интерфейса, которые будут участвовать в непосредственной обработке системного события.

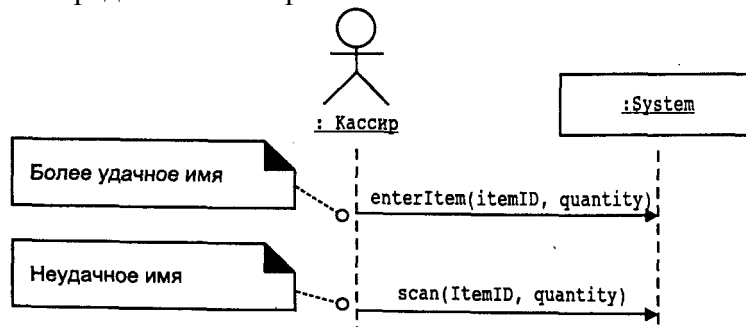


Рисунок 1.4 – Выбирайте имена событий и операций на абстрактном уровне

Отображение текста из описания прецедента

В некоторых случаях на диаграмму последовательностей следует помещать хотя бы фрагменты из описания прецедента, что позволит проиллюстрировать строгую связь между этими двумя артефактами (рис. 1.5).

Простой сценарий *Оформление продажи с оплатой наличными.*

1. Покупатель подходит к кассовому аппарату POS-системы с выбранными товарами.
2. Кассир открывает новую продажу.

3. Кассир вводит идентификатор товара.
4. Система записывает идентификатор товара и выдает его описание, цену и общую стоимость.

Кассир повторяет действия, описанные в пп. 3-4, для каждого наименования товара.

5. Система вычисляет общую стоимость покупки с налогом.

6. Кассир сообщает покупателю общую стоимость и предлагает оплатить покупку.
7. Покупатель оплачивает покупку, система обрабатывает платеж.

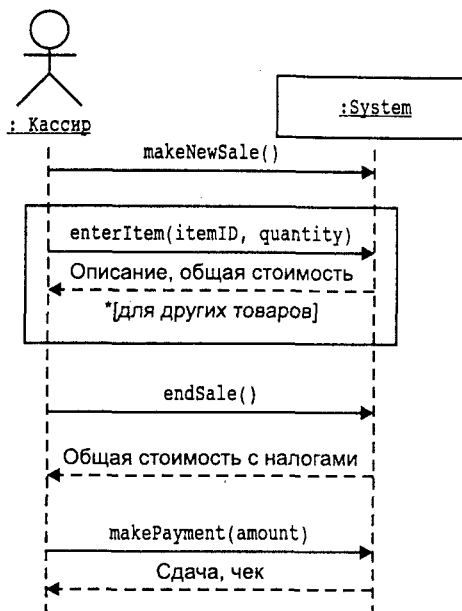


Рисунок 1.5 – Диаграмма последовательностей системы с текстом из описания прецедента

Вопрос 2. Модель прецедентов: детализация с помощью описания операций

Основным механизмом описания поведения в RUP являются прецеденты. Зачастую приводимой в них информации достаточно для описания поведения системы. Однако иногда требуется более детальное описание поведения.

Описания Системных операций (system operation contract) описывают детальное поведение системы в терминах изменения состояния объектов модели предметной области после выполнения системных операций.

Описания определяются для **системных операций** (system operations).

Системные операции – это операции, входящие в открытый интерфейс системы для обработки входных системных событий, которые система выполняет как «черный ящик». Системные операции можно идентифицировать на основе системных событий (рис. 2.1).

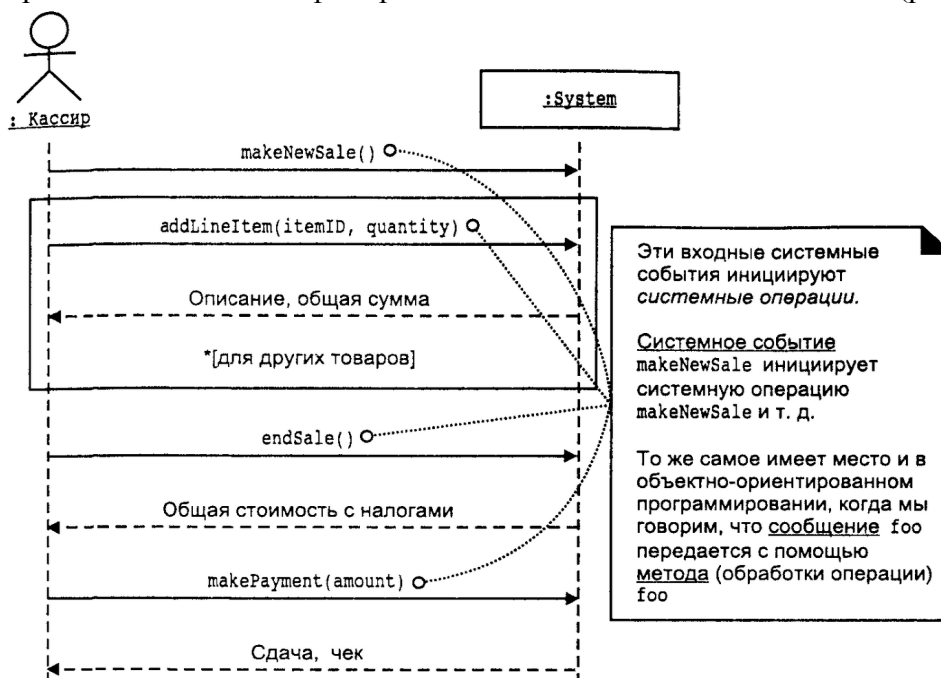


Рисунок 2.1 – Системные операции обрабатывают входные системные события

Весь набор системных операций, выполняемых в процессе всех прецедентов, определяет открытый системный интерфейс, в ракурсе которого система рассматривается как единый компонент или класс. В UML систему в целом можно представить в виде одного класса.

Разделы описания

Операция	Имя операции и ее параметры
Ссылки	(не обязательный) Прецеденты, в рамках которых может выполняться эта операция.
Предусловия	Предположения о состоянии системы или объектов модели предметной области до выполнения операции. Выполнение этих условий не проверяется в рамках логики выполнения данной операции, а предполагается, что они истинны. Это нетривиальные условия, на которые читатель должен обратить
Постусловия	Состояние объектов модели предметной области после завершения операции (подробнее обсуждается ниже).

Постусловия

В разделе "Постусловия" декларируются изменения состояния объектов модели предметной области. К таким изменениям относятся создание экземпляра, формирование или разрыв ассоциации, или изменение атрибута.

Постусловия – это не действия, выполняемые в процессе операции, а лишь декларация об изменении состоянии объектов модели предметной области после выполнения операции.

Существуют следующие категории постусловий описаний:

- создание и удаление экземпляра;
- модификация атрибута;
- формирование и разрыв ассоциаций.

Описания операций могут вообще не понадобиться. Об этом говорилось выше. Однако если нужно составить описания системных операций, то генерировать полный и детальный список постусловий на этапе анализа требований нежелательно. Их надо рассматривать как некие начальные сведения и понимать, что описания операций являются далеко не полными. Однако желательно создавать описания операций (даже неполные) на ранних этапах разработки, а не откладывать это до стадии проектирования. На стадии проектирования разработчики уже должны принимать проектные решения, а не выяснять, *что* должно быть сделано. Детали реализации операций будут исследованы на этапе проектирования.

Обсуждение: постусловия описания операции enterItem

В этом подразделе подробно рассматриваются постусловия системной операции enterItem.

Создание и удаление экземпляра

Какие новые объекты создаются в системе, когда кассир вводит код товара itemID и количество его единиц quantity? Должен быть создан экземпляр объекта SalesLineItem.

Таким образом:

- создан экземпляр sli класса SalesLineItem (*создание экземпляра*).

Обратите внимание на имя экземпляра. Это имя упростит ссылки на вновь созданный экземпляр в последующих постусловиях.

Модификация атрибута

Какие атрибуты новых или уже существующих объектов должны быть модифицированы, когда кассир вводит код товара itemID и количество его единиц quantity? Должно быть установлено количество покупаемых единиц для товара SalesLineItem.

Таким образом:

- атрибуту sli.quantity присвоено значение quantity (*модификация атрибута*).

Формирование и разрыв ассоциации

Какие ассоциации между новыми или существующими объектами должны быть сформированы или разорваны, когда кассир вводит код товара itemID и количество его единиц

quantity? Новый покупаемый товар SalesLineItem должен быть связан с текущей продажей Sale, а также со своей спецификацией productSpecification.

Таким образом:

- экземпляр sli связан с текущим экземпляром класса Sale (*формирование ассоциации*);
- экземпляр sli связан с классом ProductSpecification на основе соответствия идентификатора товара itemID (*формирование ассоциации*).

Обратите внимание на неформальную констатацию формирования взаимосвязи с конкретным экземпляром класса ProductSpecification на основе соответствия идентификатора товара itemID.

Составление описания

Чтобы составить описания для каждого прецедента, выполните следующие действия.

1. Определите системные операции из диаграмм последовательностей.
2. Составьте описание для сложных системных операций, результаты которых с очевидностью не следуют из описания прецедента.
3. При описании постусловий используйте следующие категории:
 - создание и удаление экземпляра;
 - модификация атрибута;
 - формирование и разрыв ассоциаций.

Советы по составлению описаний системных операций

- Постусловия целесообразно описывать в декларативной форме, желательно с использованием глаголов пассивного залога в прошедшем времени, чтобы подчеркнуть факт изменения состояний, а не способ реализации, как, например, показано ниже.
 - Создан экземпляр SalesLineItem (лучше).
 - Создание экземпляра SalesLineItem (хуже).
- Необходимо установить отношения между существующими и вновь создаваемыми объектами путем формирования ассоциаций. Например, при выполнении операции enterItem недостаточно просто создать новый экземпляр записи о покупке товара SalesLineItem. После выполнения операции этот экземпляр должен быть связан с текущей продажей Sale. Поэтому одним из постусловий этой операции является следующее.
 - Вновь созданный экземпляр SalesLineItem связан с объектом Sale (*формирование ассоциации*).

Пример POS-системы ТТ: описания

Системные операции для прецедента оформление продажи *Описание операции*

ОП 1: makeNewSale

Операция	makeNewSale()
Ссылки	Прецеденты: Оформление продажи
Предусловия	Отсутствуют
Постусловия	- Создан экземпляр s объекта Sale (<i>создание экземпляра</i>); - Экземпляр Sale связан с объектом Register (<i>формирование ассоциации</i>); - Инициализированы атрибуты экземпляра s.

Обратите внимание на нечеткое описание последнего постусловия. В реальном проекте все эти постусловия настолько очевидны и следуют из описания прецедентов, что создавать описание системной операции makeNewSale не имеет смысла.

Напомним один из основных принципов RUP: стремитесь к максимальной простоте и избегайте использования артефактов, которые не добавляют новых знаний.

ОП 2: enterItem

Операция	enterItem (itemID: itemID, quantity: integer)
Ссылки	Прецеденты: Оформление продажи
Предусловия	Инициирована продажа

Постусловия	<ul style="list-style-type: none"> - Создан экземпляр sli класса SalesLineItem (<i>создание экземпляра</i>); - Экземпляр sli связан с текущим экземпляром класса Sale (<i>формирование ассоциации</i>); - Атрибуту sli.quantity присвоено значение quantity (<i>модификация атрибута</i>); - Экземпляр sli связан с классом ProductSpecification на основе
-------------	--

ОП 3: endSale

Операция	endSale()
Ссылки	Прецеденты: Оформление продажи
Предусловия	Инициирована продажа
Постусловия	- Атрибут Sale.iscomplete принял значение true (<i>модификация атрибута</i>).

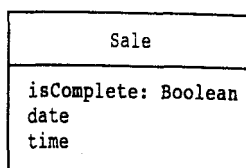
ОП 4: makePayment

Операция	makePayment (amount: Money)
Ссылки	Прецеденты: Оформление продажи
Предусловия	Инициирована продажа
Постусловия	<ul style="list-style-type: none"> - Создан экземпляр p класса Payment (<i>создание экземпляра</i>); - Атрибут p.amountTendered принял значение amount (<i>модификация атрибута</i>); - Экземпляр p связан с текущим экземпляром класса Sale (<i>формирование ассоциации</i>); - Текущий экземпляр Sale связан с экземпляром класса Store

Изменение модели предметной области

При составлении этих описаний предполагалось наличие одного объекта данных, который еще не был представлен в модели предметной области, а именно – завершение ввода информации о покупаемых товарах. Значение этого объекта изменяется при выполнении операции endSale, а в операции makePayment оно проверяется в предусловии.

Одним из способов представления этой информации является введение атрибута isComplete (или entryIsComplete) логического типа для объекта Sale.



Задание на самостоятельную работу (для выбранной темы индивидуального проекта):

1. Составить диаграмму последовательностей системы (рис. 1.5);
2. Составить описания операций, обрабатывающих входные системные события;