

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Национальный исследовательский ядерный университет «МИФИ»  
Обнинский институт атомной энергетики  
Отделение интеллектуальных кибернетических систем

**Лабораторная работа №4**

по курсу «Проектирование информационных систем»

Вариант 1.

Подписи:

Исполнитель

студент гр. ИС-Б17

\_\_\_\_\_

В. Ю. Петренко

Принял

д.т.н., профессор

\_\_\_\_\_

Н. Л. Сальников

Обнинск, 2020

**Тема:** Описание проектируемой системы с помощью UML диаграммы последовательности (sequence diagram).

Вариант 1 - Система скидок (сезонные, постоянным клиентам) в магазинах (паттерн «Стратегия»).

Диаграмма последовательности (sequence diagram) – диаграмма, на которой показаны взаимодействия объектов, упорядоченные по времени их проявления.

Для диаграммы последовательности ключевым моментом является динамика взаимодействия объектов во времени. При этом диаграмма последовательности имеет как бы два измерения. Одно – слева направо, перечисляет объекты с их линиями жизни.

Второе измерение – вертикальная временная ось, направленная сверху вниз. Каждый объект графически изображается в форме прямоугольника. Внутри прямоугольника записываются имя объекта со строчной буквы и имя класса, разделенные двоеточием. При этом вся запись подчеркивается, что является признаком объекта, который, как указывалось ранее, представляет собой экземпляр класса. Если на диаграмме последовательности отсутствует собственное имя объекта, то при этом должно быть указано имя класса. Такой объект считается анонимным. Может отсутствовать и имя класса, но при этом должно быть указано собственное имя объекта. Такой объект считается сиротой.

Крайним слева на диаграмме изображается объект-инициатор моделируемого процесса взаимодействия. Порядок расположения объектов на диаграмме последовательности определяется соображениями удобства визуализации их взаимодействия друг с другом.

Процесс взаимодействия объектов реализуется посредством сообщений, которые посылаются одними объектами другим. Сообщения изображаются в виде горизонтальных стрелок с именем сообщения. Сообщения, расположенные на диаграмме выше, передаются раньше тех, которые расположены ниже. При этом масштаб на оси времени не указывается, диаграмма моделирует лишь временную упорядоченность взаимодействий типа "раньше-позже".

Линия жизни объекта (object lifeline) – вертикальная линия на диаграмме последовательности. Линия жизни объекта изображается

пунктирной вертикальной линией. Отдельные объекты, закончив выполнение своих операций, могут быть уничтожены, чтобы освободить занимаемые ими ресурсы. Для таких объектов линия жизни обрывается в момент его уничтожения. Для обозначения момента уничтожения объекта применяется символ в форме латинской буквы "X".

В процессе функционирования объектно-ориентированных систем одни объекты могут находиться в активном состоянии, непосредственно выполняя определенные действия, или в состоянии пассивного ожидания сообщений от других объектов. Фокус управления (focus of control) – специальный символ на диаграмме последовательности, указывающий период времени, в течение которого объект выполняет некоторое действие, находясь в активном состоянии.

Периоды активности объекта могут чередоваться с периодами его пассивности или ожидания.

В отдельных случаях инициатором взаимодействия в системе может быть актер или внешний пользователь. При этом актер изображается на диаграмме последовательности самым первым объектом слева со своим фокусом управления. Актер может иметь собственное имя либо оставаться анонимным.

В отдельных случаях объект может посылать сообщения самому себе, иницируя так называемые рефлексивные сообщения. Подобные ситуации возникают, например, при обработке нажатий на клавиши клавиатуры при вводе текста в редактируемый документ, при наборе цифр номера телефона абонента.

Если в результате рефлексивного сообщения создается новый подпроцесс или нить управления, то говорят о рекурсивном или вложенном фокусе управления.

На диаграммах последовательности могут присутствовать три разновидности сообщений, каждое из которых имеет свое графическое изображение.

Первая разновидность сообщения наиболее распространена и используется для вызова процедур, выполнения операций или обозначения отдельных вложенных потоков управления. Начало этой стрелки, как правило, соприкасается с фокусом управления того объекта-клиента, который иницирует это сообщение. Конец стрелки соприкасается с линией жизни того объекта, который

принимает это сообщение и выполняет в ответ определенные действия.

Вторая разновидность сообщения используется для обозначения простого асинхронного сообщения, которое передается в произвольный момент времени. Передача такого сообщения обычно не сопровождается получением фокуса управления объектом-получателем.

Третья разновидность сообщения используется для возврата из вызова процедуры. В процедурных потоках управления эта стрелка может быть опущена, поскольку её наличие неявно предполагается в конце активизации объекта. Для непроцедурных потоков управления, включая параллельные и асинхронные сообщения, стрелка возврата должна указываться явным образом.

Каждое сообщение на диаграмме последовательности ассоциируется с определенной операцией, которая должна быть выполнена принявшим его объектом. При этом операция может иметь аргументы или параметры, значения которых влияют на получение различных результатов. Соответствующие параметры операции будет иметь и вызывающее это действие сообщение. Более того, значения параметров отдельных сообщений могут содержать условные выражения, образуя ветвление или альтернативные пути основного потока управления.

Одна из особенностей диаграммы последовательности – возможность визуализировать простое ветвление процесса. Для изображения ветвления используются две или более стрелки, выходящие из одной точки фокуса управления объекта. При этом рядом с каждой из них должно быть явно указано соответствующее условие ветви в форме булевского выражения.

Количество ветвей может быть произвольным, однако наличие ветвлений может существенно усложнить интерпретацию диаграммы последовательности. Предложение условия должно быть явно указано для каждой ветви и записывается в форме обычного текста, псевдокода или выражения языка программирования. Это выражение всегда должно возвращать некоторое булевское выражение. Запись этих условий должна исключать одновременную передачу альтернативных сообщений по двум и более ветвям.

С помощью ветвления можно изобразить и более сложную логику взаимодействия объектов между собой. Если условий более

двух, то для каждого из них необходимо предусмотреть ситуацию единственного выполнения.

На диаграммах последовательности при записи сообщений также могут использоваться стереотипы, рассмотренные при построении диаграммы кооперации. Их семантика и синтаксис остаются без изменения, как они определены в нотации языка UML.

Как уже отмечалось ранее, сообщения могут иметь собственное имя, в качестве которого выступает имя операции, вызов которой инициируют эти сообщения у принимающего объекта. В этом случае рядом со стрелкой записывается имя операции с круглыми скобками, в которых могут указываться параметры или аргументы соответствующей операции. Если параметры отсутствуют, то скобки после имени операции всё равно должны быть изображены.

Построение диаграммы последовательности целесообразно начинать с выделения из всей совокупности классов только тех, объекты которых участвуют в моделируемом взаимодействии. После этого все объекты наносятся на диаграмму, с соблюдением порядка инициализации сообщений. Здесь необходимо установить, какие объекты будут существовать постоянно, а какие временно – только на период выполнения ими требуемых действий. Когда объекты визуализированы, можно приступить к спецификации сообщений.

При этом необходимо учитывать те операции, которые имеют классы соответствующих объектов в модели системы. При необходимости уточнения этих операций следует использовать их стереотипы. Для уничтожения объектов, которые создаются на время выполнения своих действий, нужно предусмотреть явное сообщение. Наиболее простые случаи ветвления процесса взаимодействия можно изобразить на одной диаграмме с использованием соответствующих графических примитивов. В более сложных случаях для моделирования каждой ветви управления может потребоваться отдельная диаграмма последовательности.

Общим правилом является визуализация особенностей реализации каждого варианта использования на отдельной диаграмме последовательности. В этой ситуации отдельные диаграммы должны рассматриваться совместно как одна модель взаимодействия. Необходимость синхронизации сложных потоков управления, как правило, требуют введение в модель дополнительных ограничений.

## Паттерн «Стратегия»

Паттерн «Стратегия» переносит в отдельную иерархию классов все детали, связанные с реализацией алгоритмов. Абстрактный базовый класс иерархии объявляет интерфейс, общий для всех алгоритмов. Подклассы реализуют его в соответствии с тем или иным алгоритмом. Класс содержит указатель на объект абстрактного типа и предназначен для переадресации пользовательских запросов конкретному алгоритму. Для замены одного алгоритма другим достаточно перенастроить этот указатель на объект нужного типа.

### Достоинства паттерна «Стратегия»:

- систему проще поддерживать и модифицировать, так как семейство алгоритмов перенесено в отдельную иерархию классов;
- паттерн «Стратегия» предоставляет возможность замены одного алгоритма другим в процессе выполнения программы;
- паттерн «Стратегия» позволяет скрыть детали реализации алгоритмов от клиента.

### Недостатки паттерна «Стратегия»:

- для правильной настройки системы пользователь должен знать об особенностях всех алгоритмов;
- число классов в системе, построенной с применением паттерна «Стратегия», возрастает.

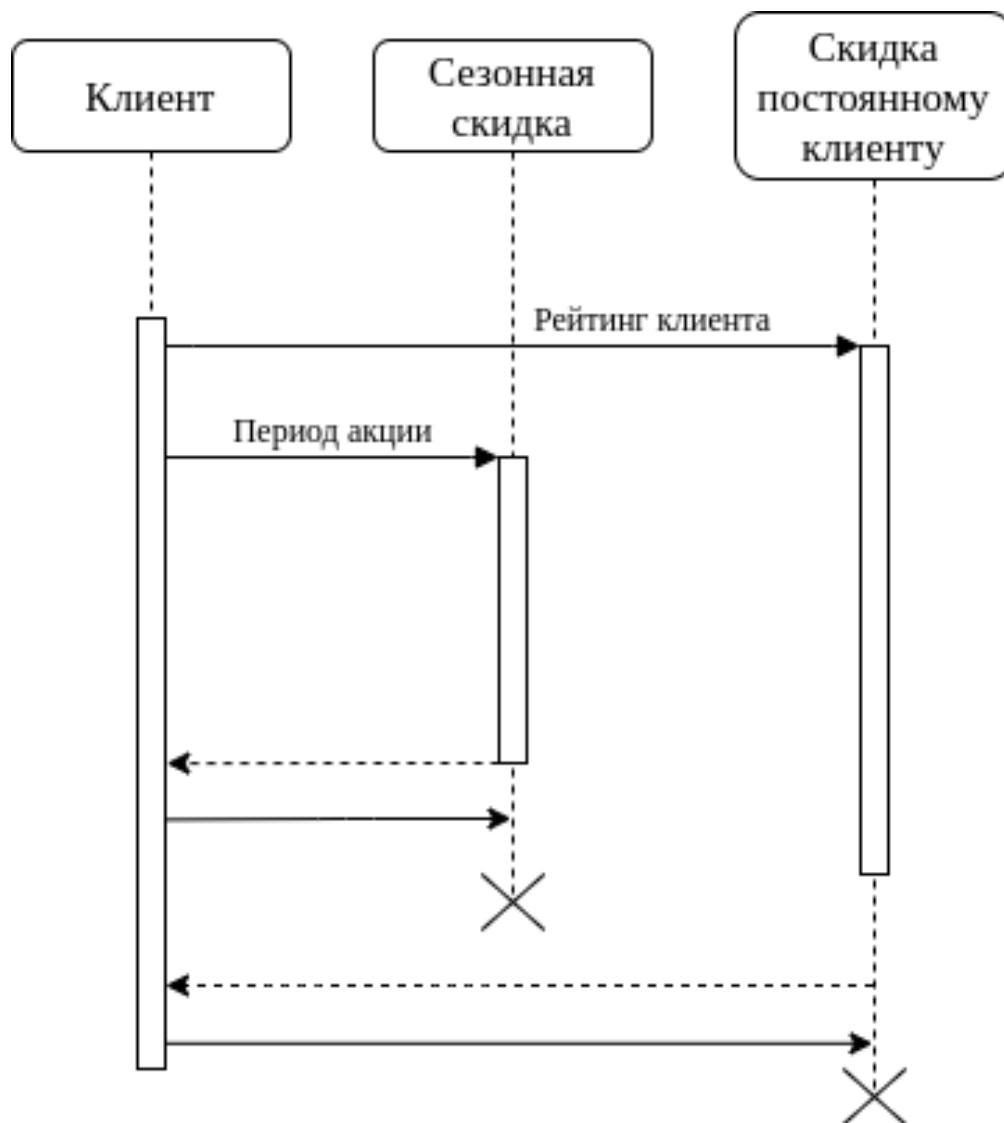


Рисунок 1 — Диаграмма последовательности системы скидок