

## **Практическое занятие** **Построение динамической модели АСОИУ.** **Диаграммы взаимодействия**

**Цель занятия:** получить навыки построения диаграмм последовательностей и кооперации.

### **Теоретические сведения к практическому занятию**

Динамическое представление логической модели – это диаграммы взаимодействия, которые используются для моделирования взаимодействия объектов.

Основным элементом диаграммы взаимодействия является объект, который описывает нечто содержащее в себе данные и поведение, объекты изображаются в виде прямоугольников, имена объектов подчеркиваются. Внутри прямоугольника, обозначающего объект, записывается с большой буквы имя объекта. Имя объекта подчеркивается. Если оно содержит несколько слов, то все они должны начинаться с большой буквы.

Для выявления объектов взаимодействия необходимо изучить имена существительные в потоке событий.

При проектировании этих диаграмм можно представлять себе объекты, как экранные формы или части приложений, отвечающие за выполнение определенных действий, или объектом может быть запись в таблице базы данных.

Существует два аспекта взаимодействия.

1. Взаимодействие объектов во времени. Для представления взаимодействия во времени используется *диаграмма последовательности*.

2. Структура взаимодействия объектов. Для представления структуры взаимодействия объектов используется *диаграмма коммуникации*.

На обеих диаграммах отображается одна и та же информация, но разными способами: диаграмма последовательностей изображает поток управления, а диаграмма коммуникации – поток данных.

Диаграммы последовательностей моделируют взаимодействия между объектами в одном варианте использования.

Диаграмма последовательности структурирована таким образом, что представляет собой временную шкалу, которая начинается сверху и постепенно спускается, отмечая последовательность взаимодействий. У каждого объекта есть столбец, а сообщения, которыми они обмениваются, представлены стрелками.

Этапы построения диаграммы последовательности:

1. нанести на диаграмму классы с соблюдением порядка вовлечения их во взаимодействие (крайний левый – актер или объект, являющийся инициатором сообщений, правее тот, который взаимодействует с первым и т.д.);

2. спецификация сообщений – сценарий событий;

### 3. нанесение сообщений на диаграмму.

Таблица 1

Элементы диаграммы

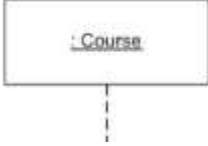




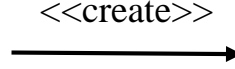

Элемент	Графическое изображение	Описание
Линия жизни объекта		Диаграмма состоит из нескольких таких обозначений линий жизни, которые должны быть расположены горизонтально в верхней части диаграммы. Никакие два обозначения жизненного пути не должны перекрывать друг друга. Они представляют различные объекты или части, которые взаимодействуют друг с другом в системе во время последовательности, существование объекта в течение определенного периода времени.
Фокус управления		Это рамка, помещенная на линию жизни. Он используется, чтобы указать, что объект активен (или создан) во время взаимодействия между двумя объектами. Длина прямоугольника указывает на то, как долго объекты остаются активными.
Стрелки сообщений		Стрелка от вызывающего сообщения к получателю сообщения указывает сообщение на диаграмме последовательности. Сообщение может течь в любом направлении; слева направо, справа налево или назад к самому сообщению. Хотя вы можете описать сообщение, отправляемое от одного объекта к другому с помощью стрелки, с помощью разных стрелок вы можете указать тип отправляемого или получаемого сообщения.

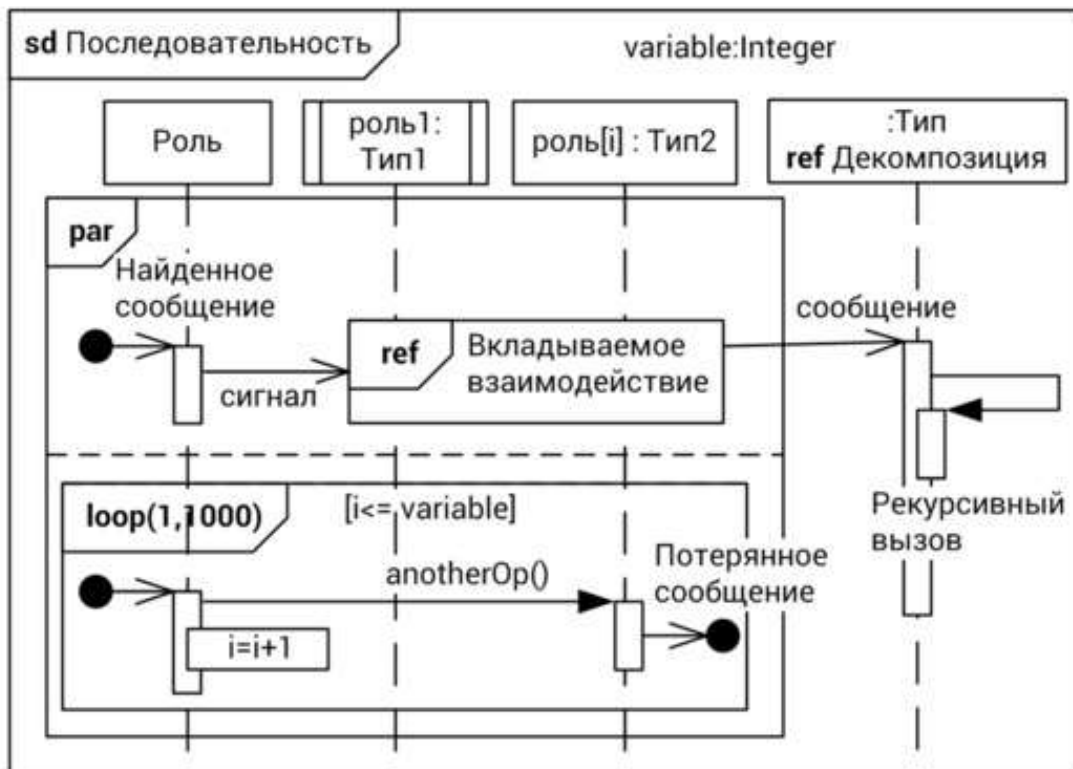
Таблица 2

Типы сообщений

Тип сообщения	Графическое изображение	Описание
Синхронное		синхронное сообщение используется, когда отправитель ожидает, пока получатель обработает сообщение и вернется, прежде чем продолжить с другим сообщением

Асинхронное		Асинхронное сообщение используется, когда вызывающий сообщение не ожидает, пока получатель обработает сообщение и вернется, прежде чем отправлять другие сообщения другим объектам в системе
Обратное		Ответное сообщение используется, чтобы указать, что получатель сообщения завершил обработку сообщения и возвращает управление вызывающей стороне сообщения.
Создать объект		Объекты не обязательно живут на протяжении всей последовательности событий. Объекты или участники могут быть созданы в соответствии с отправляемым сообщением.
Уничтожить объект		Точно так же участники, когда они больше не нужны, также могут быть удалены из диаграммы последовательности. Это делается путем добавления «X» в конце линии жизни указанного участника.

Основные элементы диаграммы последовательности представлены на рис.1



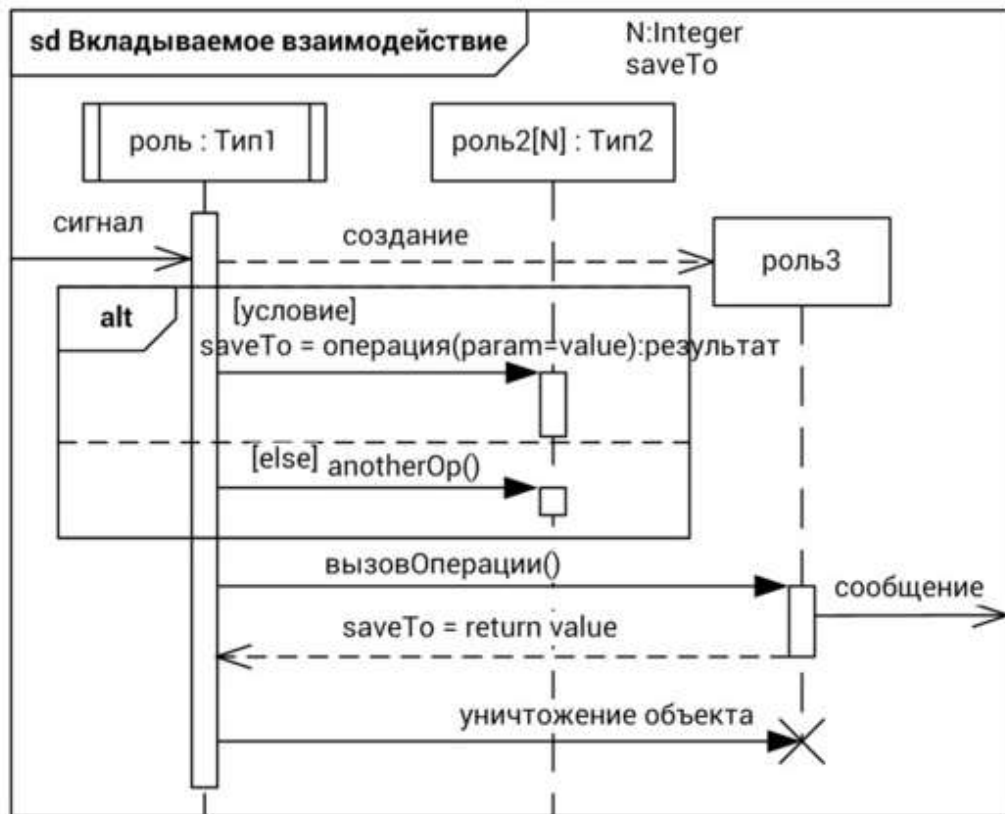


Рис. 1. Основная нотация диаграммы последовательности

Диаграммы последовательности и диаграммы коммуникации показывают одну и ту же информацию, но просто представляют ее по-разному

Диаграмма коммуникации также известная как диаграмма взаимодействия, является иллюстрацией отношений и взаимодействий между объектами. Эти диаграммы можно использовать для изображения динамического поведения конкретного варианта использования и определения роли каждого объекта.

Элементы диаграммы коммуникации:

**Объекты:** объект представлен символом объекта с подчеркнутым именем и классом, разделенными двоеточием. На диаграмме сотрудничества объекты используются следующим образом:

- Объект представлен указанием их имени и класса.
- Появление каждого класса не обязательно.
- Класс может составлять более одного объекта.
- На диаграмме сотрудничества сначала создается объект, а затем указывается его класс.
- Чтобы отличить один объект от другого, необходимо дать им имя.

**Актеры:** На диаграмме сотрудничества актер играет главную роль, поскольку он вызывает взаимодействие. У каждого актера есть соответствующая роль и имя. В этом случае один субъект инициирует вариант использования.

**Ссылки:** ссылка - это экземпляр ассоциации, которая связывает объекты и акторов. Он отображает отношения между объектами, через которые

отправляются сообщения. Он представлен сплошной линией. Ссылка помогает объекту подключиться к другому объекту или перейти к нему, так что потоки сообщений прикрепляются к ссылкам.

**Сообщения:** это сообщение между объектами, которое несет информацию и включает порядковый номер, чтобы действие могло иметь место. Он представлен стрелкой с надписью, которая размещена рядом со ссылкой. Сообщения отправляются от отправителя к получателю, и направление должно быть доступно для навигации в этом конкретном направлении. Получатель должен понимать сообщение.

Пример диаграммы коммуникации представлен на рисунке 2

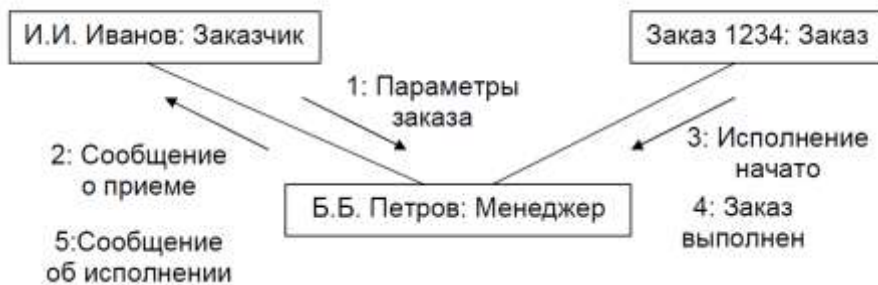


Рис. 2. Диаграмма коопераций на уровне примеров

Вариант UML-диаграммы последовательности для варианта использования *Войти в систему* приведен на рисунке 3.

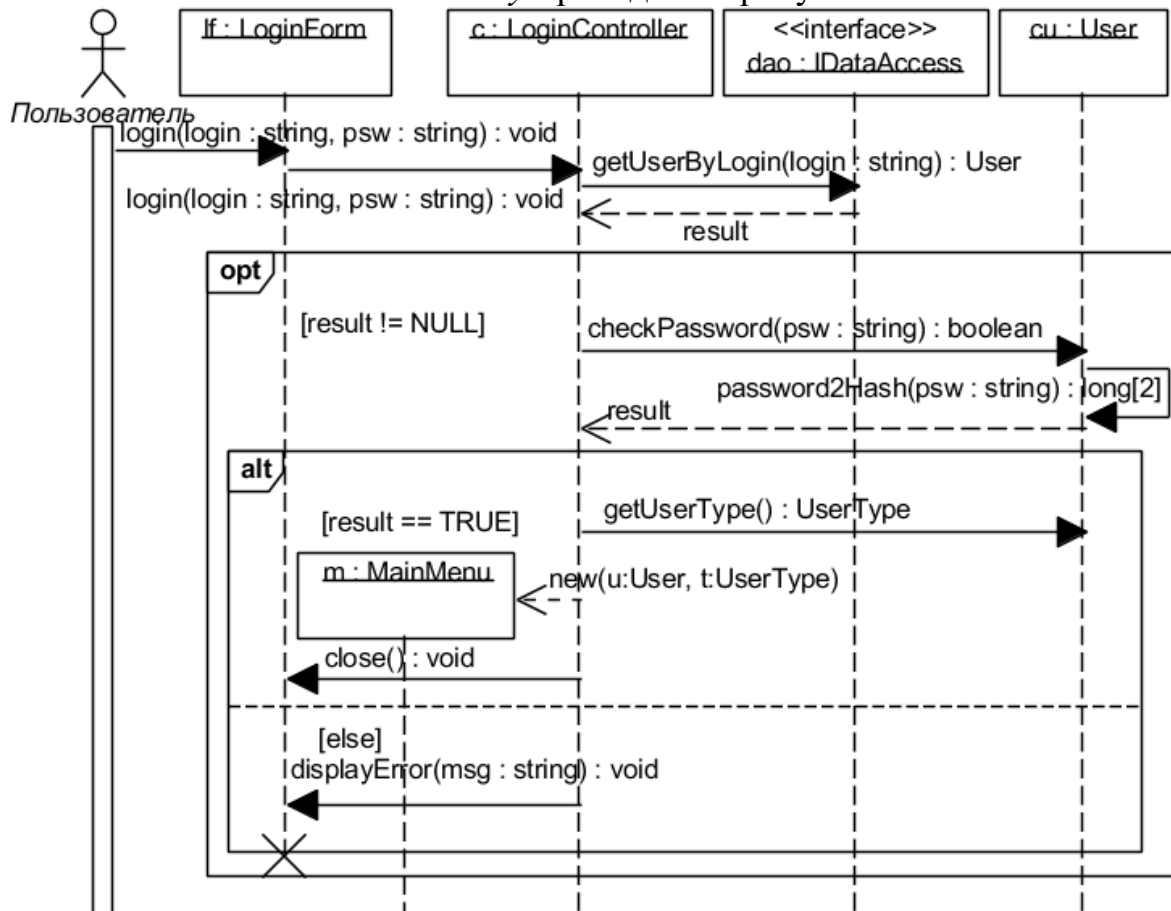


Рисунок 3 - UML-диаграмма последовательности для варианта использования *Войти в систему*

Пример диаграммы последовательности представлен на рисунке 4.

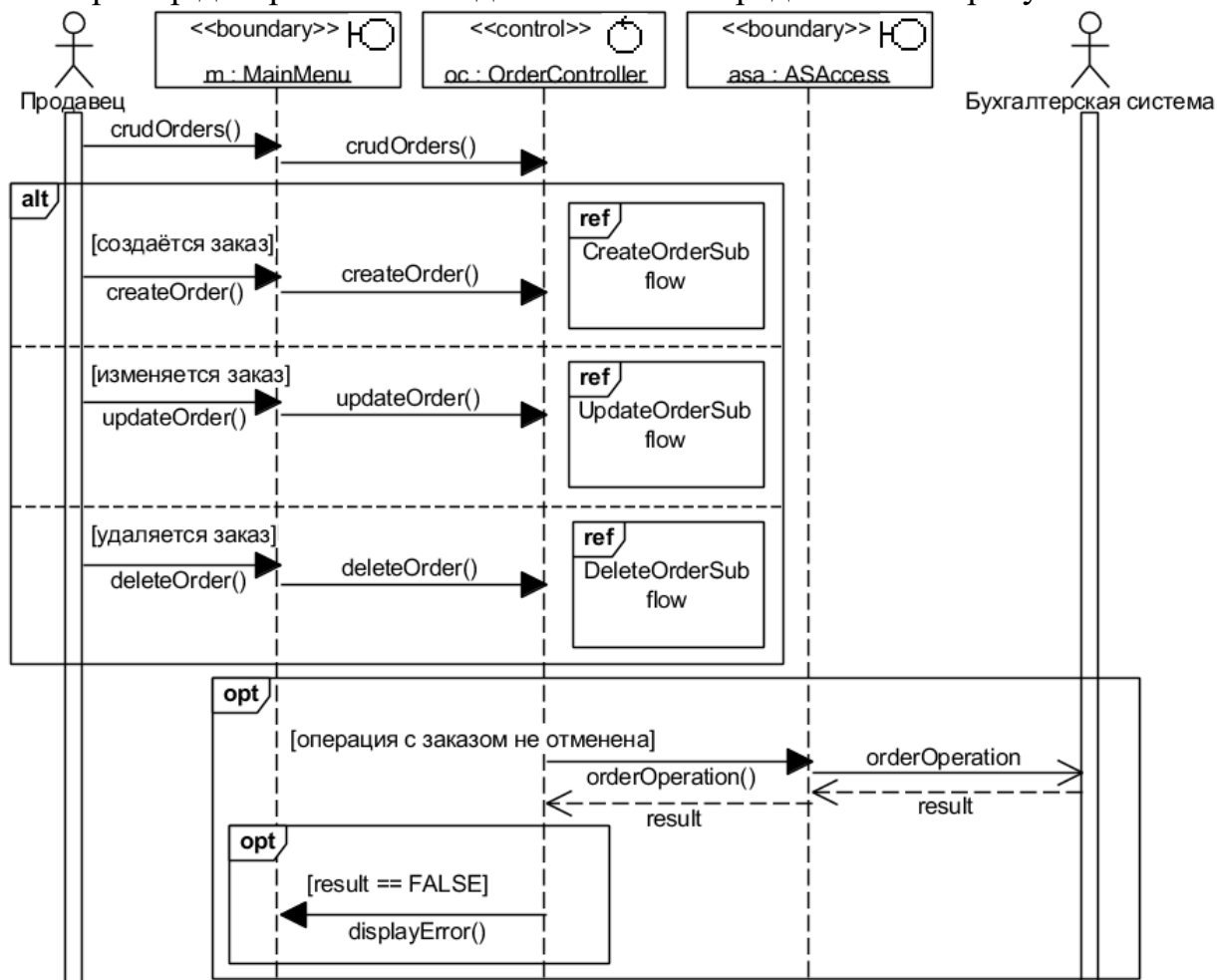


Рис. 4. UML-диаграмма последовательности CRUDOrdersBasicFlow

### Задачи для самостоятельной работы

**9.1.** Разработать диаграмму последовательностей для сценария: учитель выбирает в главном меню пункт «добавить ученика»; система показывает учителю окно добавления ученика, содержащее поля для ввода логина и пароля, а также кнопки «далее» и «назад»; учитель вводит желаемый логин и пароль ученика, нажимает кнопку «далее»; система добавляет ученика; учителю открывается главное меню и в течении 5 секунд выводится уведомление о том, что ученик был добавлен успешно.

**9.2.** Разработать диаграмму последовательностей для сценария Решение задачи (фрагмент варианта использования Выполнение задания от момента инициализации пользователем процесса решения до его завершения).

Рассмотреть два варианта последовательности действий:

- а) нормальный процесс;
- б) прерывание процесса пользователем;

Нормальный процесс предполагает, что при выдаче команды *Создать* создается объект *Решение*, управляющий данным сценарием. Следующее сообщение *Начать* активизирует этот объект. Объект *Решение* запрашивает у

объекта класса *Задание* тип объекта *Алгоритм*, создает объект требуемого класса и активизирует его, сохраняя способность получать и обрабатывать сообщения (параллельный процесс).

Объект класса *Алгоритм*, реализующий метод, запрашивает у объекта класса *Задание* данные и начинает обработку, используя вспомогательные объекты. Нормально завершив обработку, объект класса *Алгоритм*, реализующий метод, передает объекту класса *Задание* результаты и возвращает объекту *Решение* признак нормального завершения. Объект *Решение* уничтожает объект класса *Алгоритм*, реализующий метод, и возвращает вызвавшему его объекту признак нормального завершения решения. В случае прерывания процесса объект *Решение* прерывает процесс решения, уничтожает объект *Алгоритм* и возвращает признак прерванного выполнения.

**9.3.** Разработать диаграмму последовательностей для сценария: сотрудник пишет заявление о включении его в график отпусков по данному подразделению. Линейный руководитель все заявления от сотрудников и заполняет документ «предложения по графику отпусков». Менеджер по персоналу рассматривает-отклоняет, просит, уточнить, просит пересмотреть или принимает. Когда набирается большое количество заявлений, то менеджер направляет руководителю. Руководитель перепроверяет, подписывает и передает офис- менеджеру для оформления приказа.

**9.4.** Разработать диаграмму коммуникации для сценария: сначала пользователь выбирает пункт меню «Создать сеть», после чего главная форма приложения создаёт объект «Форма создания сети» и отображает эту форму на экране. С помощью неё пользователь может вводить параметры сети – длину входного вектора и число правил. После задания всех параметров пользователь может нажать на кнопку «Создать сеть», после чего форма создания сети создаст объект «Нейронная сеть». Созданная сеть будет отображена на главной форме, которая закроет и уничтожит форму создания сети.

**9.5.** Разработать диаграмму коммуникации для сценария: пользователь входит в главное окно программы и нажимает на кнопку «Отзыв заявки». Открывается окно «Отзыв заявки». Пользователь вводит номер заявки, номер паспорта и код с картинки. После чего нажимает кнопку «Готово». Система проверяет код введенный с картинки с внутренним кодом, если код введен верно, то сравниваются введенный номер паспорта с имеющимся в БД, и если они совпадают – удаляет заявку из «БД Заявок», после чего происходит переход в главное окно программы.

**9.6.** Разработать диаграмму последовательностей прецедента из простой системы обработки заказов (табл. 3), имеющей следующие ограничения:

- заказ должен полностью оплачиваться единовременным платежом;
- позиции, указанные в заказе, могут поставляться только после оплаты;
- позиции доставляются покупателю в течение 28 дней после оплаты.

## Прецедент «Сделать заказ»

Прецедент Сделать заказ
ID: 5
Краткое описание: Покупатель делает заказ, который затем оплачивается и доставляется
Главные актеры: Покупатель
Второстепенные актеры: Нет
Предусловия: Нет
Основной поток: <ol style="list-style-type: none"> <li>1. Прецедент начинается, когда актер Покупатель создает новый заказ.</li> <li>2. Покупатель полностью оплачивает заказ.</li> <li>3. Товары доставляются Покупателю в течение 28 дней после даты окончательного платежа</li> </ol>
Постусловия: <ol style="list-style-type: none"> <li>1. Заказ оплачен.</li> <li>2. Товары доставлены в течение 28 дней после окончательного платежа</li> </ol>
Альтернативные потоки: Переплата по заказу Отмена заказа Товары не поставлены Поставка с опозданием Частичная оплата