



Delivering Excellence in Software Engineering



Диаграмма последовательностей

Общие сведения

Курс включает в себя следующие презентации:

- **UML. Введение**
- **Диаграмма вариантов использования**
- **Диаграмма деятельности**
- **Диаграмма последовательности**
- **Диаграмма классов**
- **Диаграмма состояний**

Дать общие представления о диаграммах последовательностей

1 Диаграмма последовательностей

2 Элементы диаграммы последовательностей

3 Примеры

1 Диаграмма последовательностей

2 Элементы диаграммы последовательностей

3 Примеры

В основе любой сложной **системы** находятся **объекты**. В крупных системах **объекты взаимодействуют** друг с другом.

Взаимодействием (**interaction**) называется поведение, выражаемое в обмене сообщениями между объектами данной совокупности в данном контексте, в результате чего достигается определенная цель.

Для описания совместного функционирования объектов существуют диаграммы взаимодействия (**interaction diagrams**) двух видов:

- диаграмма последовательностей взаимодействия (sequence diagram)
- диаграмма сотрудничества объектов (collaboration diagram)

Диаграмма последовательностей (sequence diagram) - это диаграмма, на которой показано множество объектов и сообщений, которыми они обмениваются

Назначение диаграммы последовательностей

- **Диаграммы последовательностей служат для моделирования временной упорядоченности потоков управления**
- **Диаграммы последовательностей показывают, в каком порядке посылаются сообщения между объектами системы**

Диаграммы последовательностей могут существовать автономно, и могут использоваться для моделирования отдельного потока управления в составе прецедента

Особенности диаграммы последовательностей

- **На диаграмме последовательностей внимание акцентируется на временном порядке отправляемых и принимаемых сообщений**

Диаграмма последовательностей может содержать

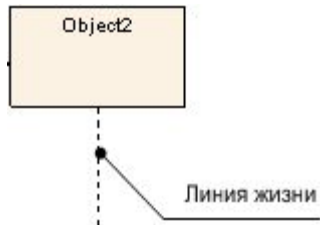
- **Объекты**
- **Сообщения**
- **Примечания**

1 Диаграмма последовательностей

2 Элементы диаграммы последовательностей

3 Примеры

Линия жизни и фокус управления



Линия жизни объекта (lifeline) - это вертикальная пунктирная линия, отражающая существование объекта во времени. Большая часть объектов существует на протяжении всего взаимодействия. Конец существования объекта показывается знаком «X» на его линии жизни



Фокус управления - вытянутый прямоугольник, показывающий промежуток времени, в течение которого объект выполняет какое-либо действие, непосредственно или с помощью подчиненной процедуры.

Начало фокуса управления выравнивается по временной оси с моментом начала действия, конец - с моментом его завершения. Вложенность фокуса управления, вызванную рекурсией или обратным вызовом со стороны другого объекта, показывается расположением другого фокуса управления чуть правее своего родителя.

Объекты

Объект (object) - конкретная материализация класса. Объекты - это экземпляры классов; можно сказать, что все объекты являются экземплярами, но не все экземпляры - объектами.

Объекты могут быть **именованные** и **анонимные**

Экземпляр (instance) - конкретная материализация абстракции, к которой могут быть применены операции и которая может сохранять их результаты.

Понятия «экземпляр» и «объект» практически синонимичны.

На диаграмме последовательностей объекты располагаются вдоль оси X

Виды объектов

Actor – экземпляр участника процесса (актера)

Lifeline – объект общего назначения

Boundary – экран пользовательского интерфейса или устройство ввода-вывода

Entity – постоянный элемент. Как правило, соответствует таблице или элементу базы данных

Control – активный элемент, который управляет выполнением процесса

Сообщения

Сообщение (message) - спецификация обмена данными между объектами, при котором передается некая информация в расчете на то, что в ответ последует определенное действие.

Получение объектом экземпляра сообщения можно считать экземпляром события.

Результатом получения сообщения является некое действие, которое может привести к изменению состояния объекта.

На диаграмме последовательностей сообщения располагаются вдоль оси Y

Виды сообщений

Вызвать (call) – вызывает операцию применимую к объекту. Объект может посылать сообщение самому себе (**recursive call**).

Возвратить (return) – возвращает значение вызывающему объекту

Послать (send) – посылает объекту сигнал

Создать (create) – создает новый объект

Уничтожить (destroy) – удаляет объект. Объект может удалять сам себя.

Виды сообщений

Синхронное сообщение

После вызова операции вызывающий объект (отправитель) ждет пока принимающий объект (получатель) примет сообщение, выполнит операцию и вернет некоторый результат.

Асинхронное сообщение

После вызова операции отправитель продолжает свое выполнение. Получатель принимает сообщение как только будет к этому готов. Отреагировав на вызов получатель продолжает работу.

Сигнал (signal) – именованный объект, который асинхронно возбуждается отправителем и перехватывается получателем

Советы

При моделировании взаимодействий соблюдайте следующие правила:

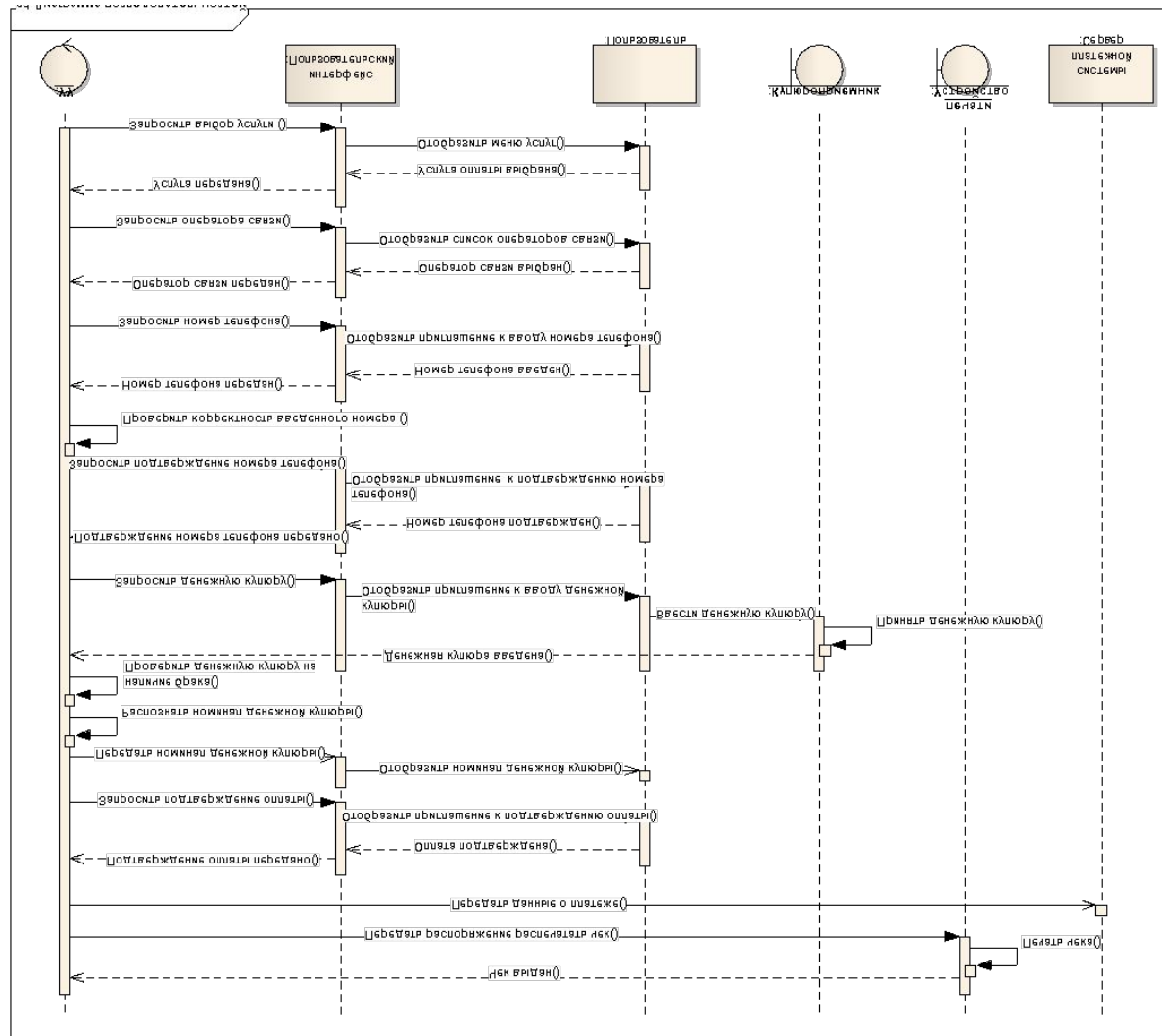
- Акцентируйте внимание только на одном аспекте динамики системы
- Включайте только тех актеров, которые важны для понимания этого аспекта
- Включайте детали, которые соответствуют данному уровню абстракции и необходимы для понимания системы
- Элементы необходимо располагать так, чтобы свести к минимуму число пересечений
- Располагайте рядом семантически близкие элементы

1 Диаграмма последовательностей

2 Элементы диаграммы последовательностей

3 Примеры

Примеры



Web-сайты

UML® Resource Page

www.uml.org

UML Tutorial

<http://www.sparxsystems.com.au/uml-tutorial.html>

UML 2.1 Tutorial

http://www.sparxsystems.com.au/resources/uml2_tutorial/

Книги

Г. Буч, Д. Рамбо, А. Джекобсон “Язык UML Руководство пользователя”

Леоненков А. “Самоучитель UML”

Martin Fowler “UML Distilled: A Brief Guide to the Standard Object Modeling Language”



Delivering Excellence in Software Engineering

Диаграмма последовательностей

For more information, please contact:

Sergey Kalinov
Software Engineering Team Leader

EPAM Systems, Inc.
Ryazan, Russia (GMT+3)

Phone: +7 920 635 02 13

Email: Sergey.Kalinov@epam.com

<http://www.epam.com>

