CASE технологии

Лекция 5

Язык UML: виды диаграмм

- UML 1.5 определял двенадцать типов диаграмм, разделенных на три группы:
  - четыре типа диаграмм представляют статическую структуру приложения;
  - пять представляют поведенческие аспекты системы;
  - три представляют физические аспекты функционирования системы (диаграммы реализации).
- Текущая версия UML 2.1 внесла не слишком много изменений. Диаграммы слегка изменились внешне (появились фреймы и другие визуальные улучшения), немного усовершенствовалась нотация, некоторые диаграммы получили новые наименования.

Язык UML: виды диаграмм

- диаграмма прецедентов;
- диаграмма классов;
- диаграмма объектов;
- диаграмма последовательностей;
- диаграмма взаимодействия;
- диаграмма состояний;
- диаграмма активности;
- диаграмма развертывания.

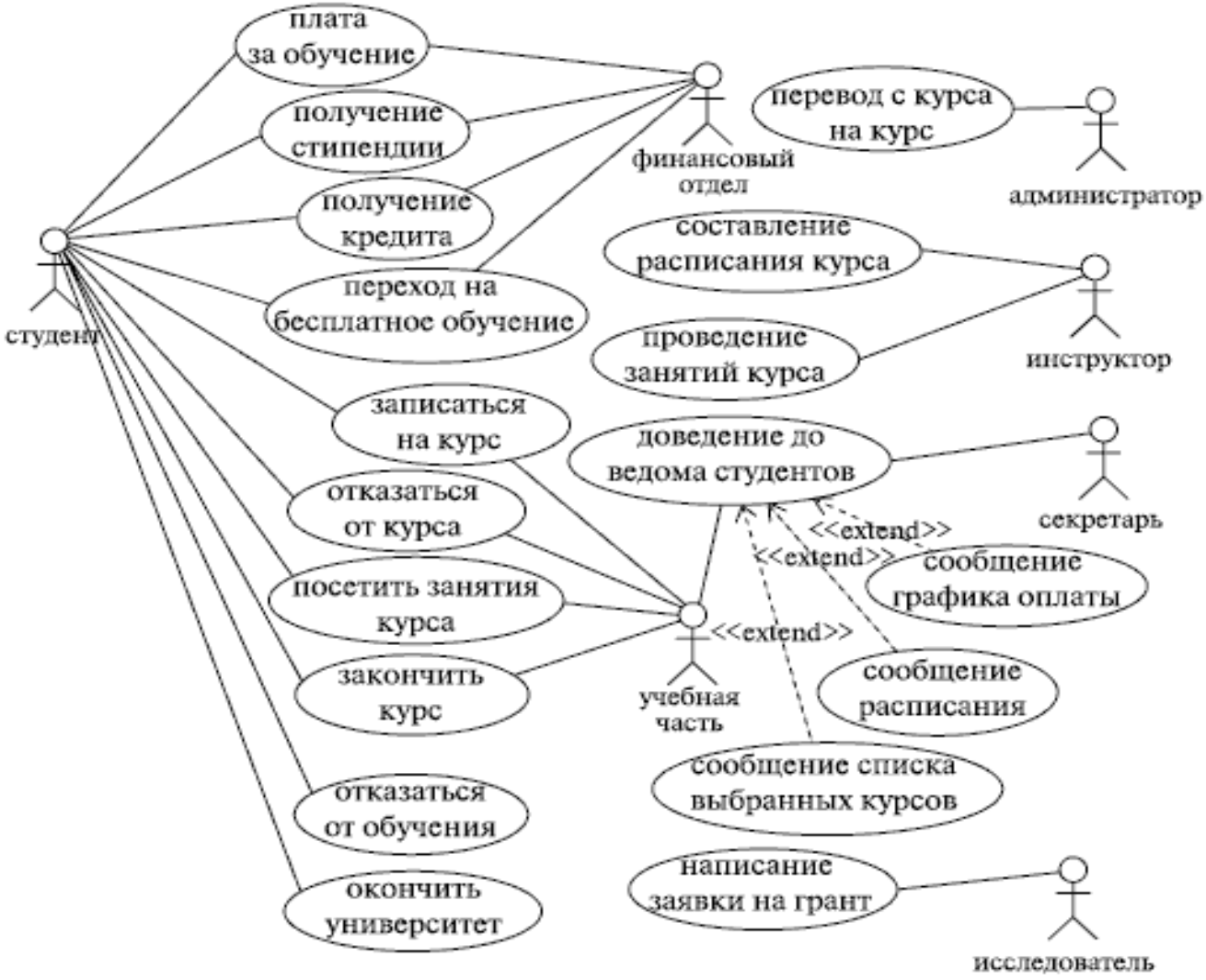
Диаграммы прецедентов (use case diagram)

- Любые (в том числе и программные) системы проектируются с учетом того, что в процессе своей работы они будут использоваться людьми и/или взаимодействовать с другими системами.
- Сущности, с которыми взаимодействует система в процессе своей работы, называются **экторами**, причем каждый эктор ожидает, что система будет вести себя строго определенным, предсказуемым образом.

Диаграммы прецедентов (use case diagram)

- **Эктор (actor)** - это множество логически связанных ролей, исполняемых при взаимодействии с прецедентами или сущностями (система, подсистема или класс).
- Эктором может быть человек или другая система, подсистема или класс, которые представляют нечто вне сущности.
- **Прецедент (use-case)** - описание отдельного аспекта поведения системы с точки зрения пользователя (Буч).

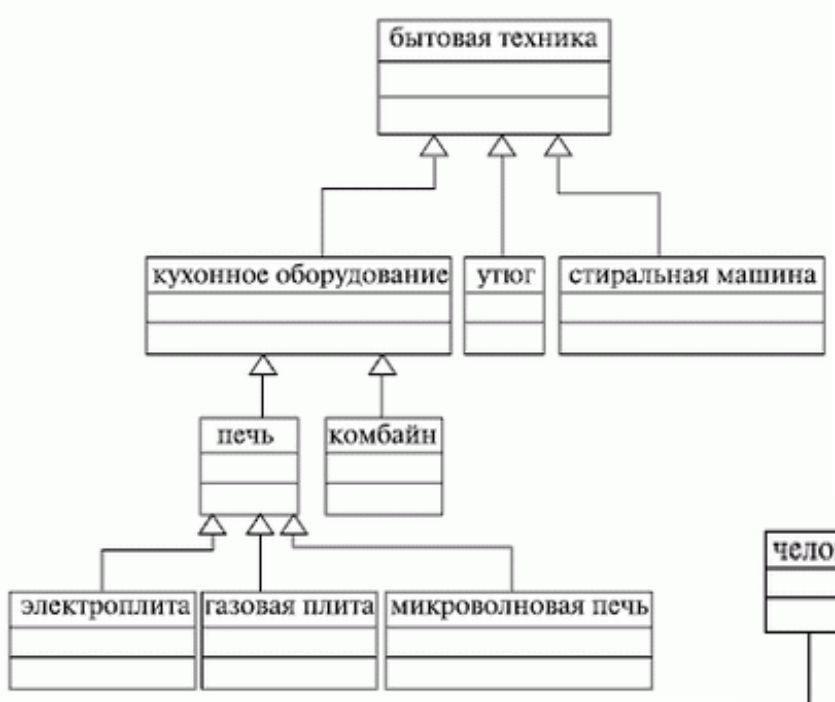
Диаграммы прецедентов



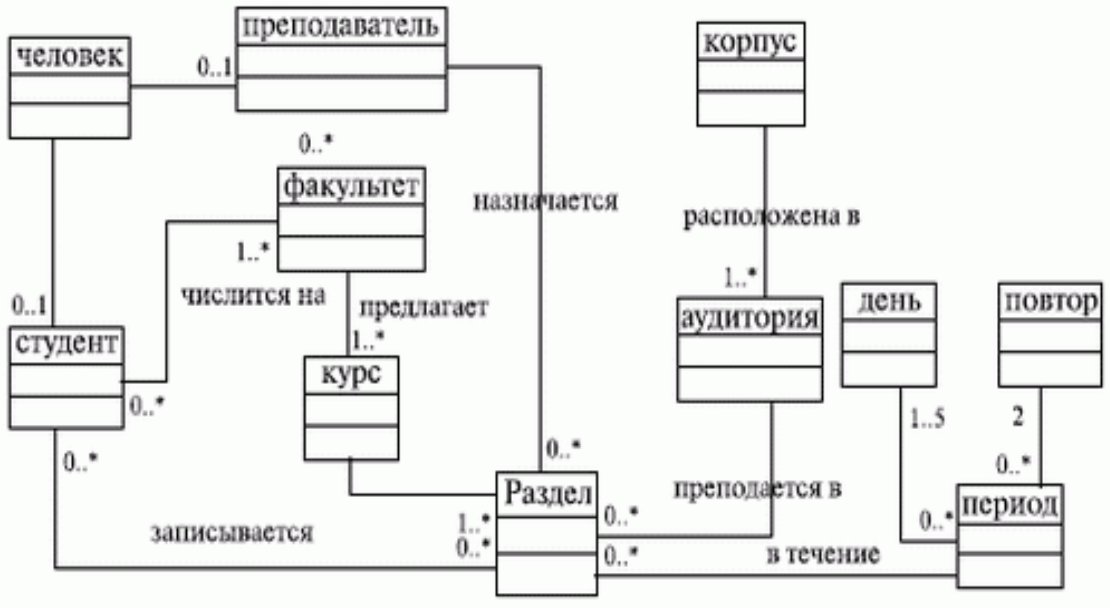
Диаграммы классов

- **Класс (class)** - категория вещей, которые имеют общие атрибуты и операции.
- Классы используются в процессе анализа предметной области для составления словаря предметной области разрабатываемой системы.
- Это могут быть как абстрактные понятия предметной области, так и классы, на которые опирается разработка и которые описывают программные или аппаратные сущности.
- **Диаграмма классов** - это набор статических, декларативных элементов модели.
- Диаграммы классов могут применяться и при прямом проектировании, то есть в процессе разработки новой системы, и при обратном проектировании - описании существующих и используемых систем.
- **Информация с диаграммы классов напрямую отображается в исходный код приложения**
- Таким образом, диаграмма классов - конечный результат проектирования и отправная точка процесса разработки.

Диаграммы классов

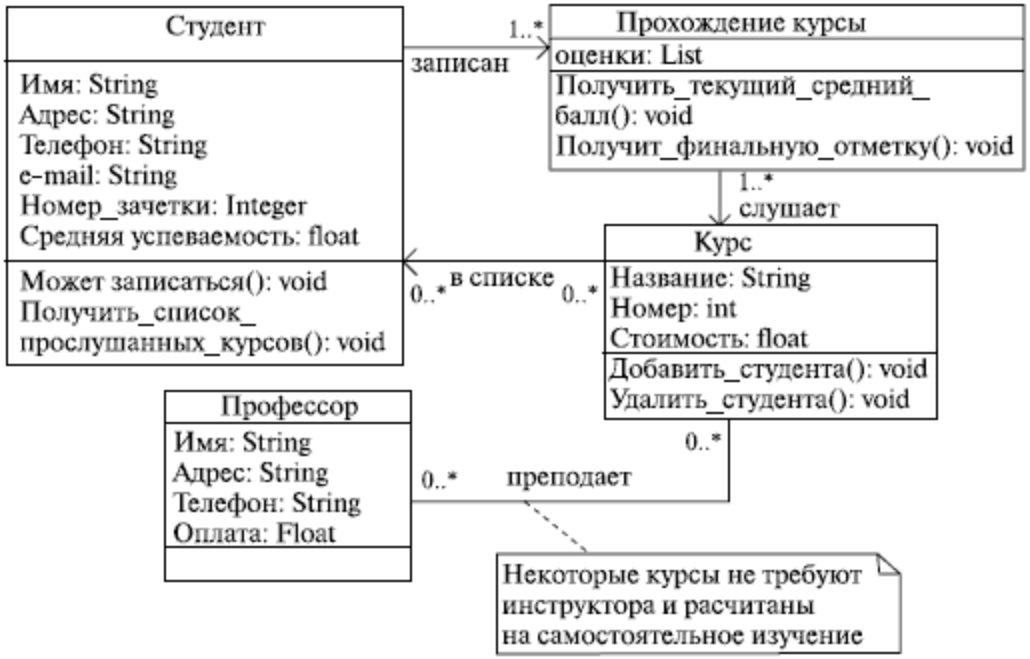


Иерархия классов



Взаимодействие между классами

Диаграммы классов



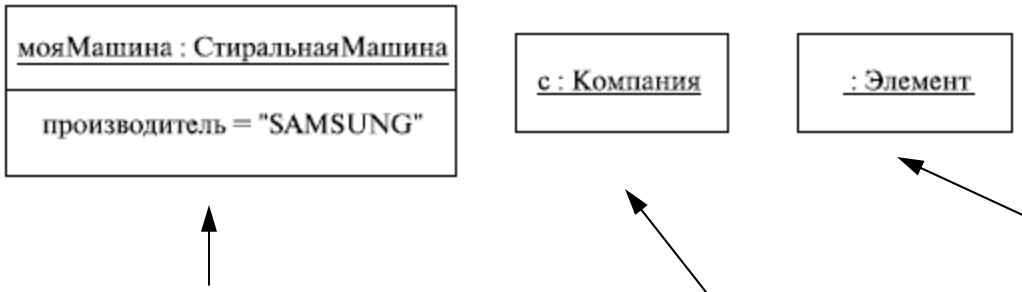
Свойства и методы классов

UML – язык для Объектно-Ориентированного Проектирования

Диаграммы объектов

- **Объект (object)** - экземпляр класса.
- Объект, как и класс, на диаграмме обозначается прямоугольником, но его имя подчеркивается.

Диаграммы объектов

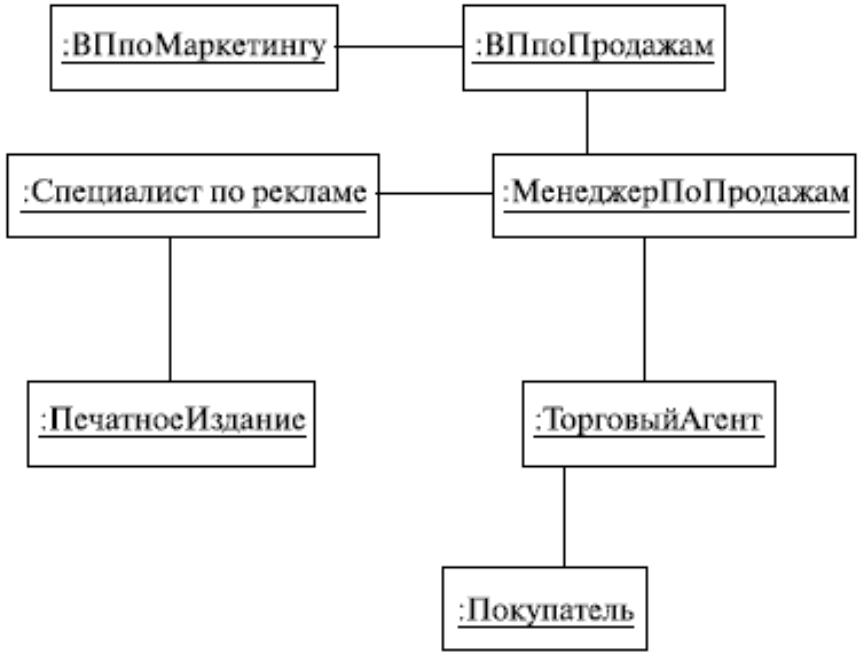


Элемент класса «Стиральная машина» определяется по значению свойства «производитель»

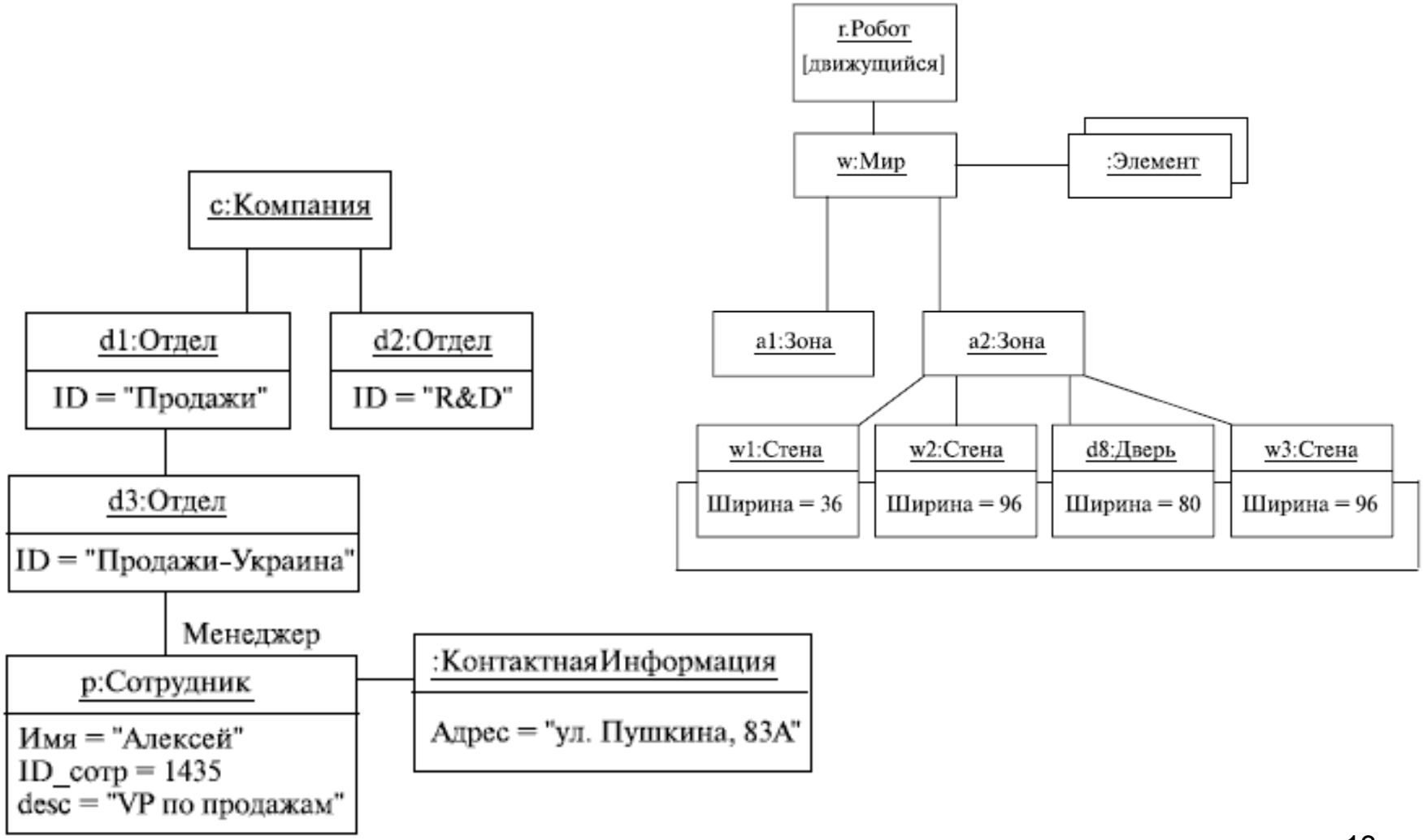
Элемент класса «Компания»

Анонимный элемент

Диаграммы объектов



Диаграммы объектов



Диаграммы последовательностей (sequence diagram)

- Диаграмма объектов показывает отношения между объектами в некоторый момент времени, т. е. предоставляет нам снимок состояния системы, являясь статической.
- Диаграмма последовательностей отображает взаимодействие объектов в динамике.

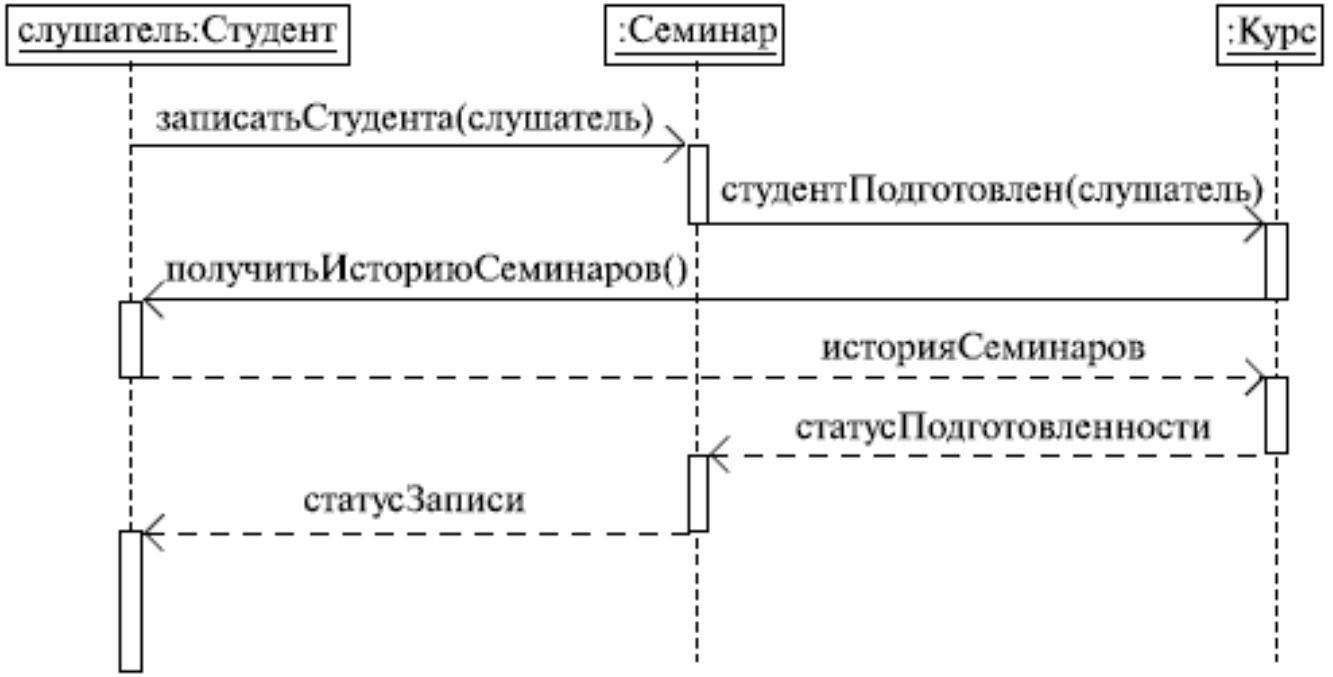
Диаграммы последовательностей (sequence diagram)

- В UML взаимодействие объектов понимается как обмен информацией между ними.
- При этом информация принимает вид сообщений.
- Кроме того, что сообщение несет какую-то информацию, оно некоторым образом также влияет на получателя.

Диаграммы последовательностей (sequence diagram)

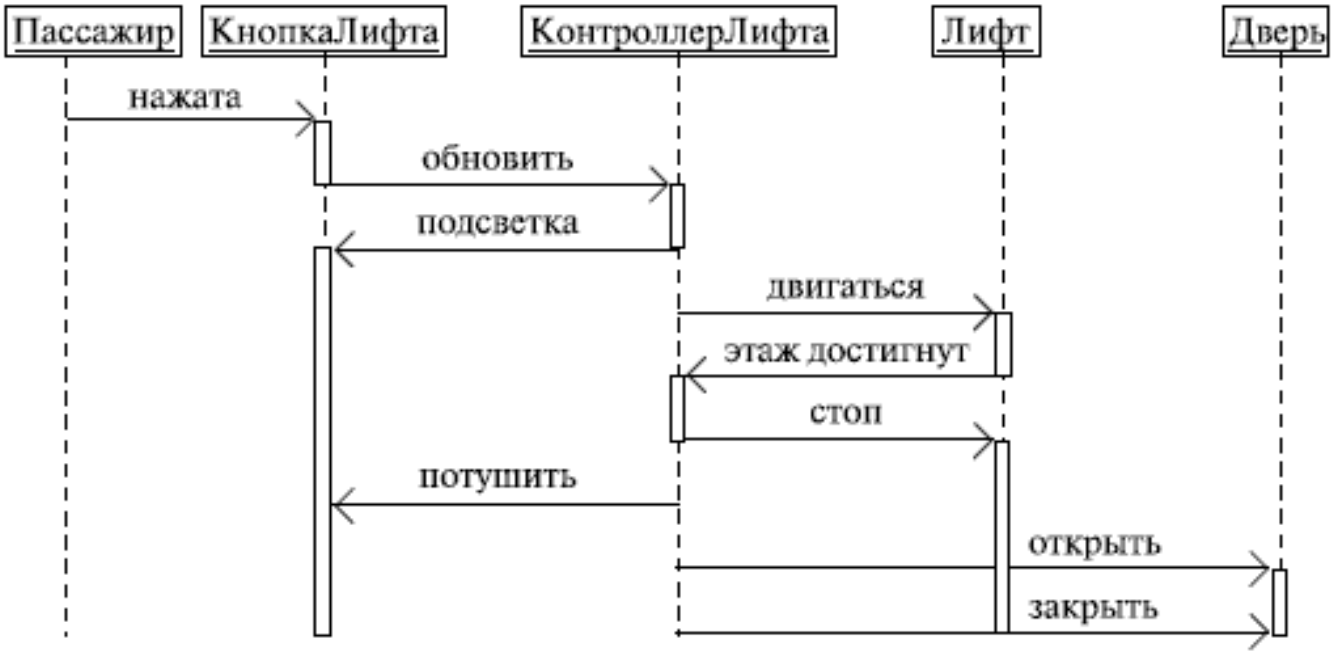
- Диаграммы последовательностей можно (и нужно!) использовать для уточнения диаграмм прецедентов
- Диаграммы последовательностей обычно содержат объекты, которые взаимодействуют в рамках сценария, сообщения, которыми они обмениваются, и возвращаемые результаты, связанные с сообщениями.

Диаграммы последовательностей

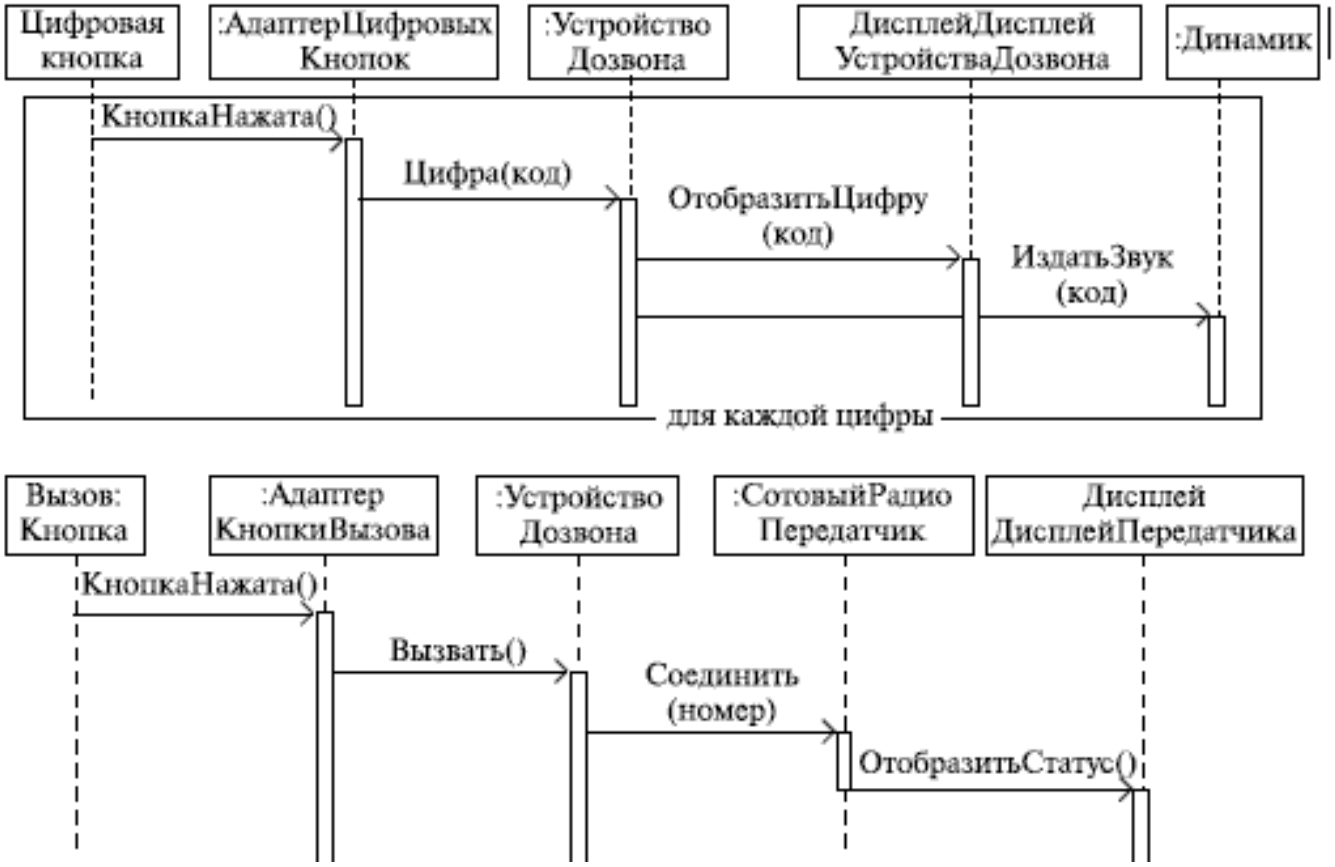


Диаграммы последовательностей

Диаграмма последовательностей отображает взаимодействие объектов в динамике



Диаграммы последовательностей



Диаграммы взаимодействия (кооперации, collaboration diagrams)

Диаграмма взаимодействия показывает поток сообщений между объектами системы и основные ассоциации между ними.

Является альтернативой диаграммы последовательностей.

Диаграммы взаимодействия (кооперации, collaboration diagrams)

Диаграммы взаимодействия (кооперации, collaboration diagrams)

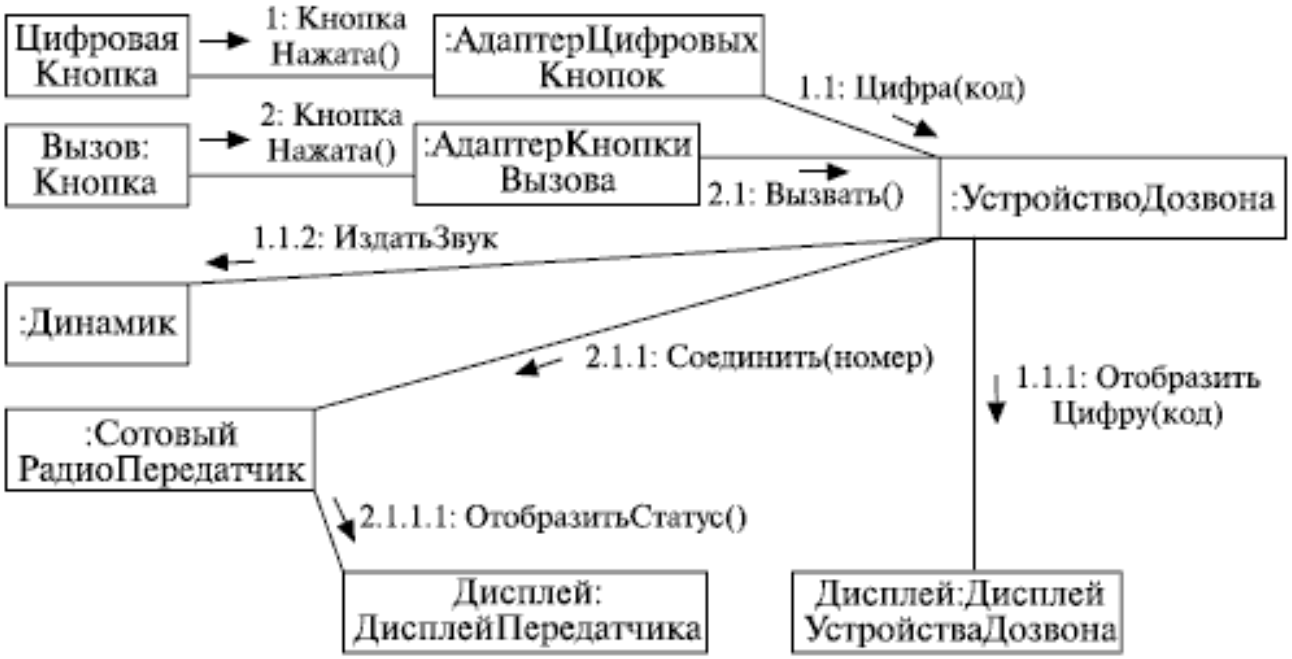


Диаграмма состояний (statechart diagram)

- Объекты характеризуются поведением и состоянием, в котором находятся. Например, человек может быть новорожденным, младенцем, ребенком, подростком или взрослым.
- Другими словами, объекты что-то делают и что-то "знают". Диаграммы состояний применяются для того, чтобы объяснить, каким образом работают сложные объекты.

Диаграмма состояний (statechart diagram)

- Состояние (state) - ситуация в жизненном цикле объекта, во время которой он удовлетворяет некоторому условию, выполняет определенную деятельность или ожидает какого-то события.
- Состояние объекта определяется значениями некоторых его атрибутов и присутствием или отсутствием связей с другими объектами.

Диаграмма состояний(statechart diagram)

- Диаграмма состояний показывает, как объект переходит из одного состояния в другое.
- Диаграммы состояний служат для моделирования динамических аспектов системы.
- Диаграмма состояний полезна при моделировании жизненного цикла объекта (как и ее частная разновидность - диаграмма деятельности)

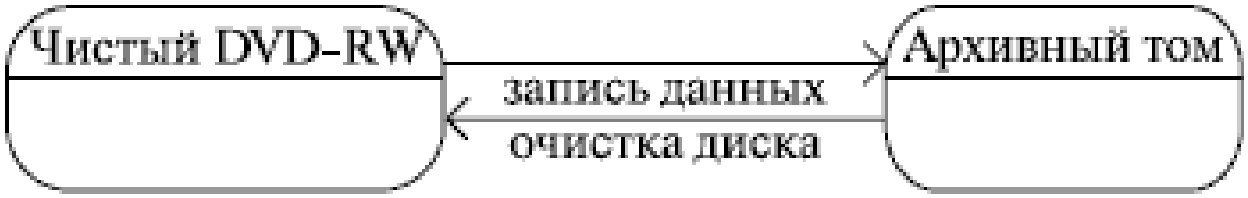


Диаграмма состояний(statechart diagram)

Составное состояние, включающее другие состояния, одно из которых содержит также параллельные подсостояния

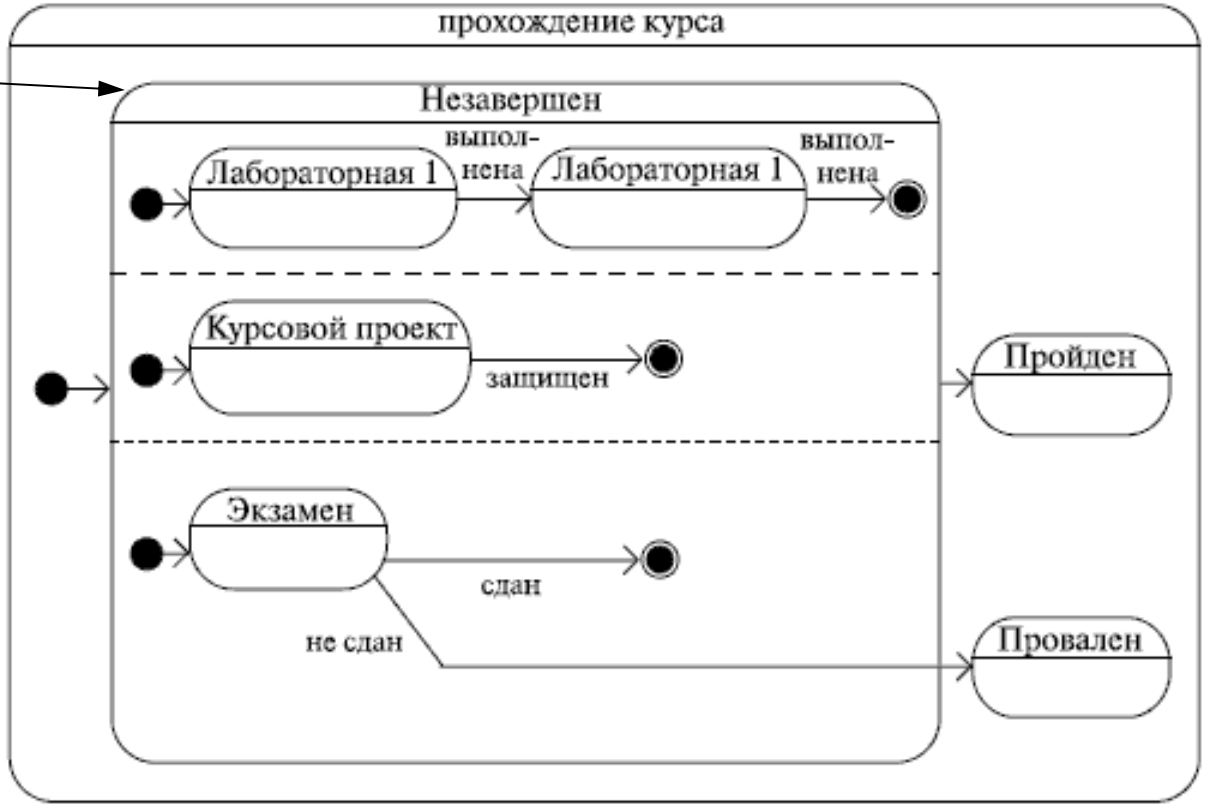


Диаграмма состояний(statechart diagram)

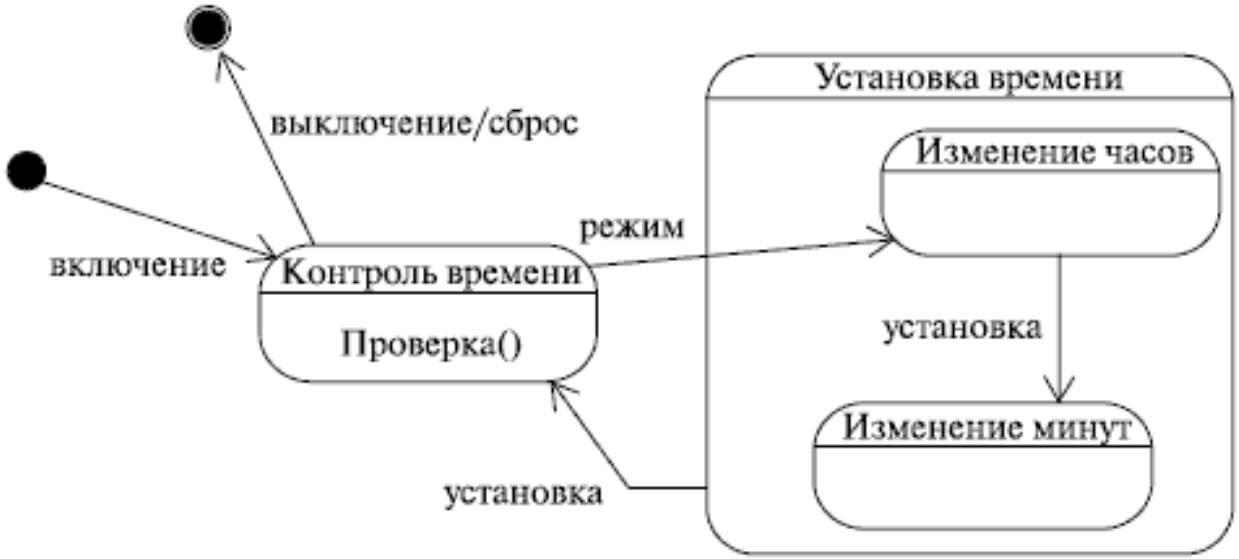
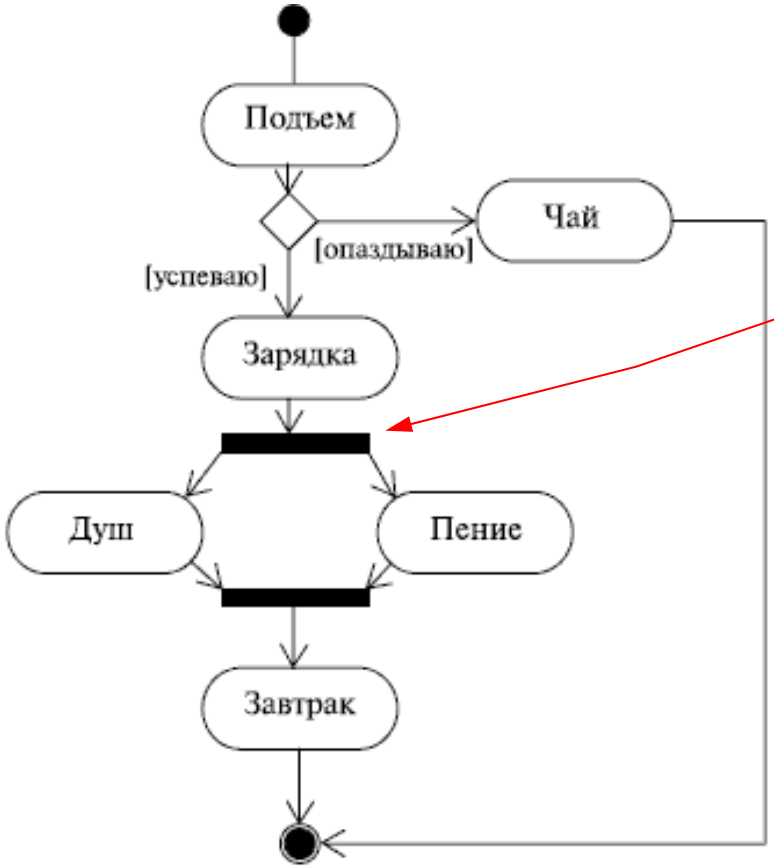


Диаграмма активности (деятельности, activity diagram)

- Моделируя поведение проектируемой системы, часто недостаточно изобразить процесс смены ее состояний, а нужно также раскрыть детали алгоритмической реализации операций, выполняемых системой.
- Для этой цели традиционно использовались блок-схемы или структурные схемы алгоритмов.
- В UML для этого существуют **диаграммы деятельности, являющиеся частным случаем диаграмм состояний**.
- Диаграммы деятельности удобно применять для визуализации алгоритмов, по которым работают операции классов.

Диаграмма активности (деятельности, activity diagram)



Параллельное исполнение

Диаграмма активности (деятельности, activity diagram)

Диаграмма активности (деятельности, activity diagram)

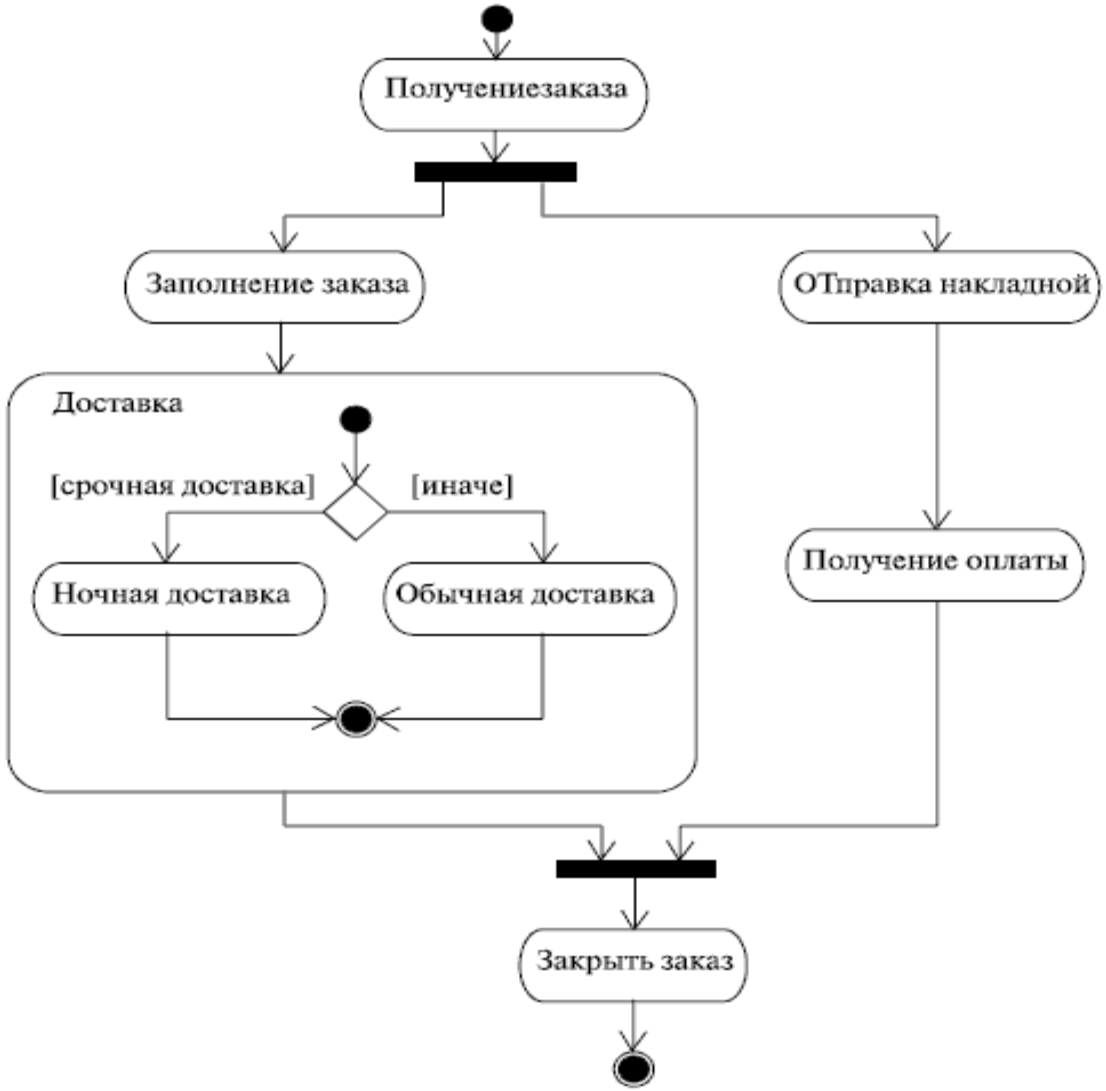


Диаграмма развертывания (deployment diagram)

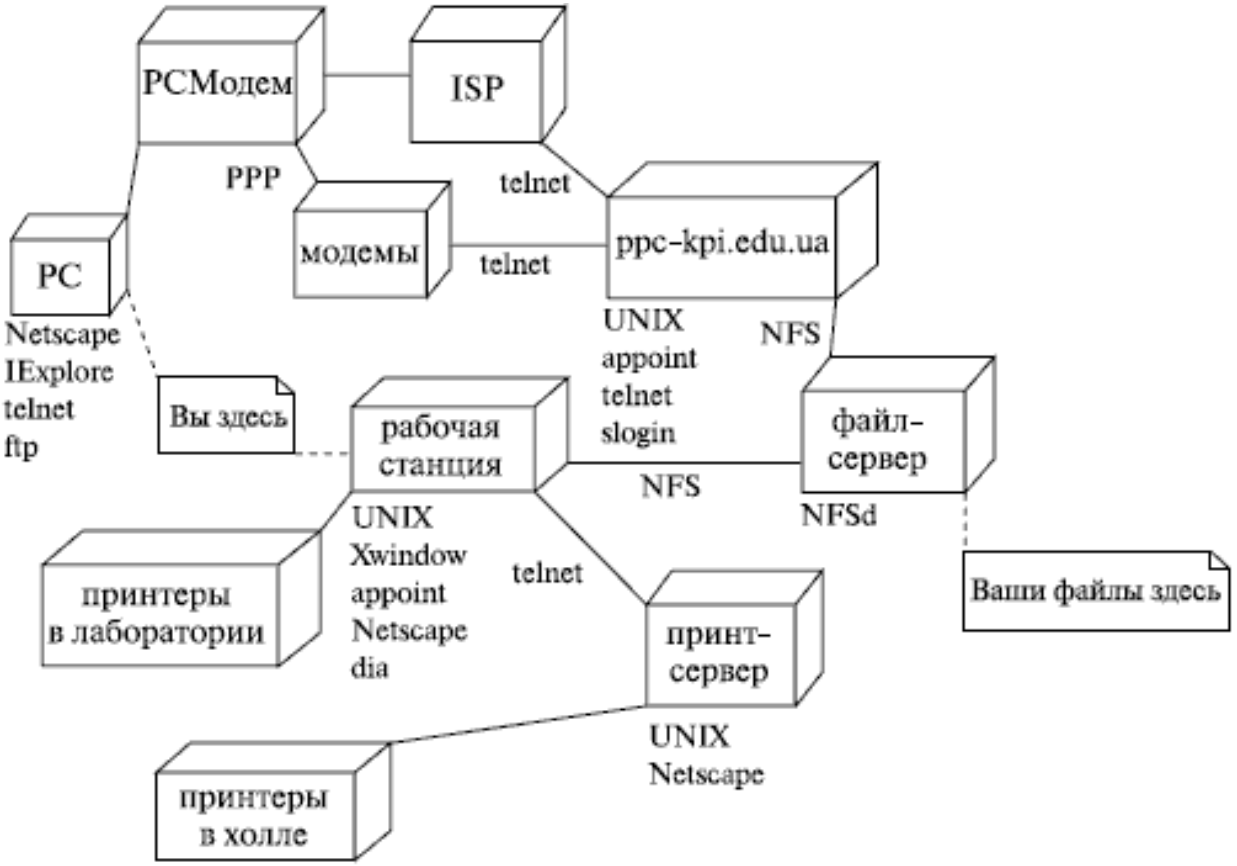
- Когда мы пишем программу, мы пишем ее для того, чтобы запускать на компьютере, который имеет некоторую аппаратную конфигурацию и работает под управлением некоторой операционной системы.
- Корпоративные приложения часто требуют для своей работы некоторой ИТ-инфраструктуры, хранят информацию в базах данных, расположенных где-то на серверах компании, вызывают веб-сервисы, используют общие ресурсы и т. д.
- В таких случаях хорошо, если будет графическое представление инфраструктуры, на которую будет развернуто приложение.
- Для этого нужны **диаграммы развертывания**, которые иногда называют **диаграммами размещения**.

Диаграмма развертывания (deployment diagram)

- Графическое представление ИТ-инфраструктуры может помочь более рационально распределить компоненты системы по узлам сети, от чего, как известно, зависит в том числе и производительность системы.
- Такая диаграмма может помочь решить множество вспомогательных задач, связанных, например, с обеспечением безопасности.
- Диаграмма развертывания показывает топологию системы и распределение компонентов системы по ее узлам, а также соединения - маршруты передачи информации между аппаратными узлами.
- Это единственная диаграмма, на которой применяются "трехмерные" обозначения: узлы системы обозначаются кубиками. Все остальные обозначения в UML - плоские фигуры.

Диаграмма развертывания (deployment diagram)

Диаграмма развертывания (deployment diagram)



Последовательность построения диаграмм

Можно предложить такую последовательность построения диаграмм:

- диаграмма прецедентов,
- диаграмма классов,
- диаграмма объектов,
- диаграмма последовательностей,
- диаграмма кооперации,
- диаграмма состояний,
- диаграмма активности,
- диаграмма развертывания.

Резюме: что узнали на лекции?

- Познакомились со всеми типами диаграмм UML