СТАТИЧЕСКИЕ МОДЕЛИ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ПРОГРАММНЫХ СИСТЕМ

Статические модели

- Статические модели обеспечивают представление структуры систем в терминах базовых строительных блоков и отношений между ними. «Статичность» этих моделей состоит в том, что здесь не показывается динамика изменений системы во времени. Вместе с тем следует понимать, что эти модели несут в себе не только структурные описания, но и описания операций, реализующих заданное поведение системы.
- Основным средством для представления статических моделей являются диаграммы классов. Вершины диаграмм классов нагружены классами, а дуги (ребра) — отношениями между ними. Диаграммы используются:
- в ходе анализа — для указания ролей и обязанностей сущностей, которые обеспечивают поведение системы;
- в ходе проектирования — для фиксации структуры классов, которые формируют системную архитектуру.
-

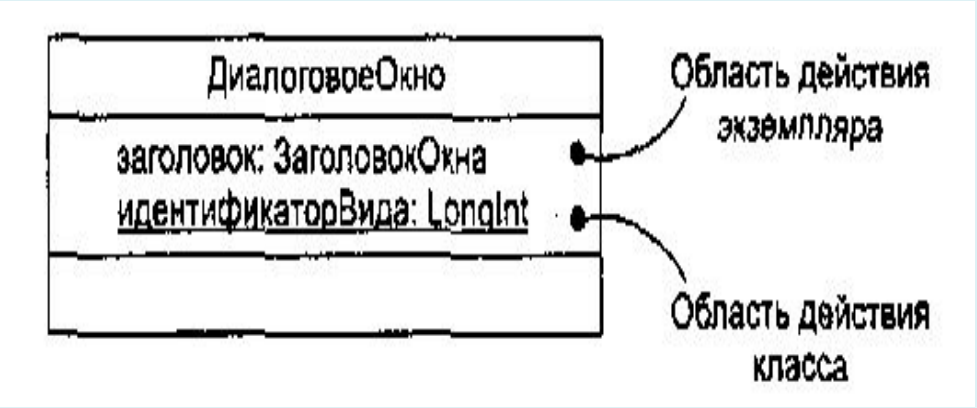
Вершины в диаграммах классов

- Итак, вершина в диаграмме классов — класс. Обозначение класса показанс

- **Рис.** Обозначение класса

- Имя класса указывается всегда, свойства и операции — выборочно.

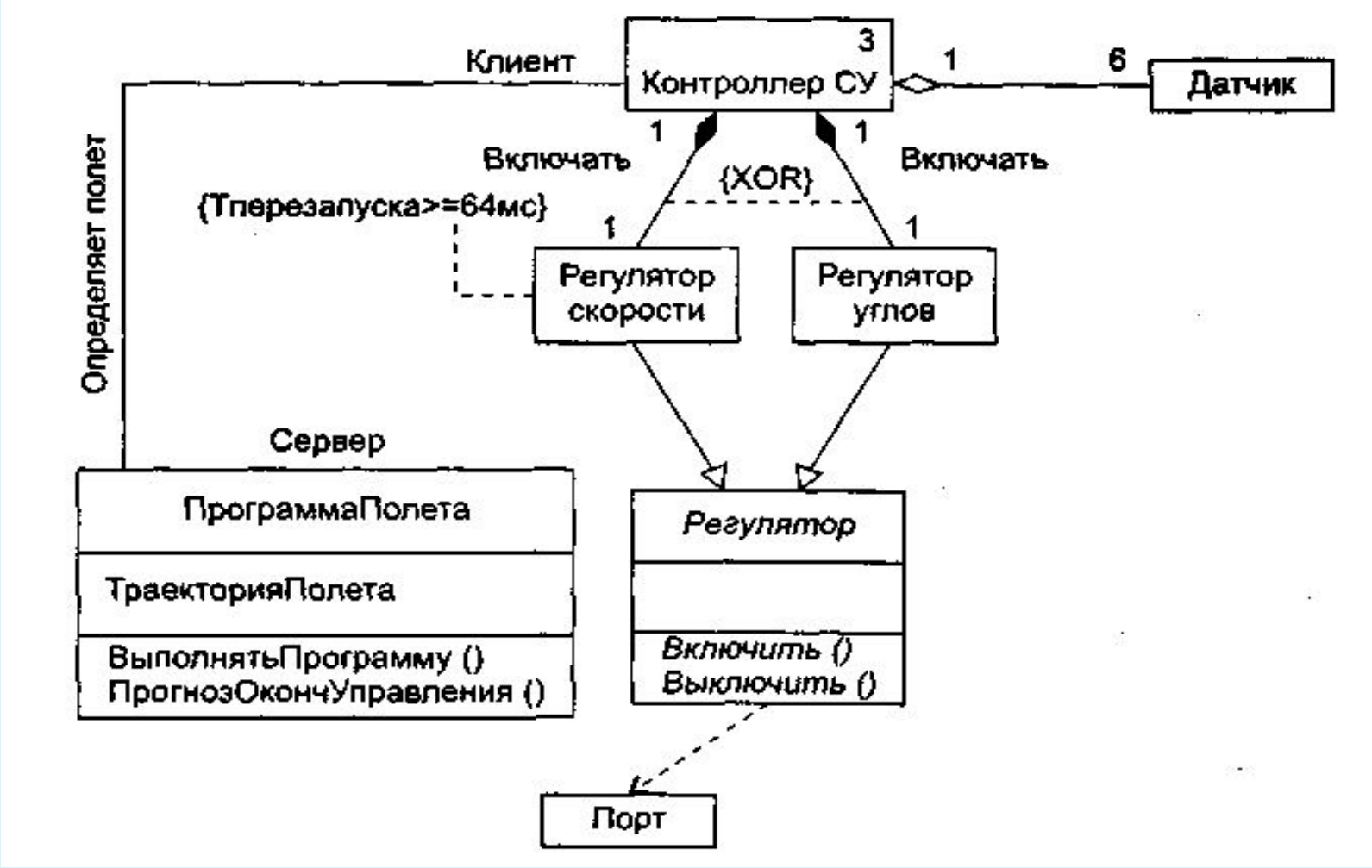
- Предусмотрено задание области действия свойства (операции). Если свойство (операция) подчеркивается, его областью действия является класс, в противном случае областью Действия является экземпляр (рис. 2).
- Что это значит? Если областью действия свойства является класс, то все его экземпляры (объекты) используют общее значение этого свойства, в противном случае у каждого экземпляра свое значение свойства.



- **Рис. 2.** Свой

Примеры диаграмм классов

- В качестве первого примера показана диаграмма классов системы управления полетом летательного аппарата.



диаграмм классов

-
- Здесь представлен класс ПрограммаПолета, который имеет свойство ТраекторияПолета, операцию-модификатор ВыполнятьПрограмму () и операцию-селектор ПрогнозОкончУправления (). Имеется ассоциация между этим классом и классом Контроллер СУ — экземпляры программы задают параметры движения, которые должны обеспечивать экземпляры контроллера.
- Класс Контроллер СУ — агрегат, чьи экземпляры включают по одному экземпляру классов Регулятор скорости и Регулятор углов, а также по шесть экземпляров класса Датчик. Экземпляры Регулятора скорости и Регулятора углов включены в агрегат физически (с помощью отношения *композиция),* а экземпляры Датчика — по ссылке, то есть экземпляр Контроллера СУ включает лишь указатели на объекты-датчики. Регулятор скорости и Регулятор углов — это подклассы абстрактного суперкласса *Регулятор,* который передает им в наследство абстрактные операции *Включить ( )* и *Выключить ().* В свою очередь, класс *Регулятор* использует конкретный класс Порт.

диаграмм классов

- Как видим, ассоциация имеет имя (Определяет полет), роли участников ассоциации явно указаны (Сервер, Клиент). Отношения композиции также имеют имена (Включать), причем на эти отношения наложено ограничение — контроллер не может включать Регулятор скорости и Регулятор углов одновременно.
- Для класса Контроллер СУ задано ограничение на множественность — допускается не более трех экземпляров этого класса. Класс Регулятор скорости имеет ограничение другого типа — повторное включение его экземпляра разрешается не раньше, чем через 64 мс.

диаграмма классов информационной системы театра

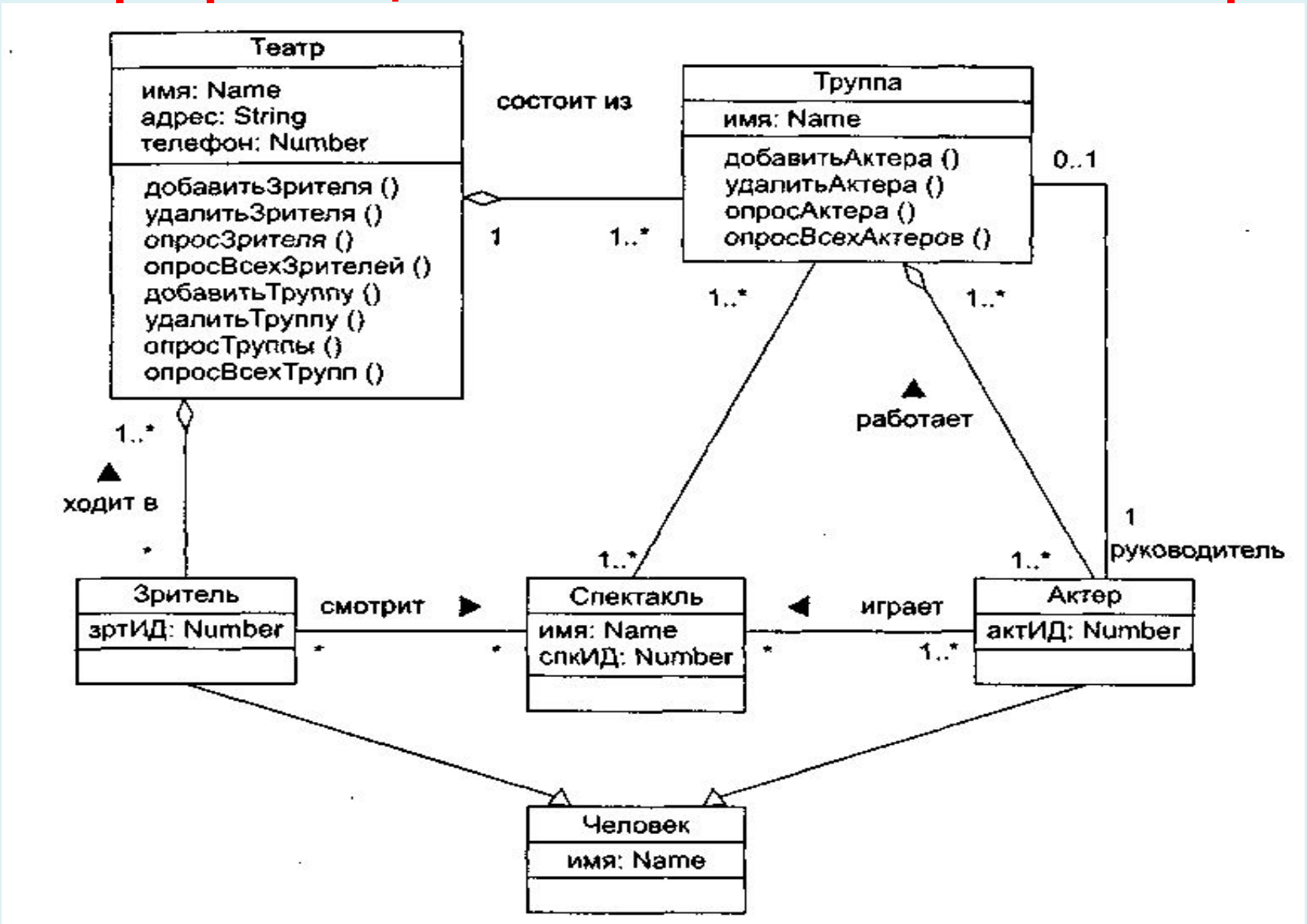


диаграмма классов информационной системы театра

- Эту систему образует 6 классов.
- Классы-агрегаты Театр и Труппа имеют операции добавления и удаления своих частей, которые включаются в агрегаты по ссылке. Частями Театра являются Зрители и Труппы, а частями Труппы — Актеры. Отношения агрегации между классом Театр и классами Труппа и Зритель слегка отличны. Театр может состоять из одной или нескольких трупп, но каждая труппа находится в одном и только одном театре. С другой стороны, в театр может ходить любое количество зрителей (включая нулевое количество), причем зритель может посещать один или несколько театров.
- Между классами Труппа и Актер существуют два отношения — агрегация и ассоциация. Агрегация показывает, что каждый актер работает в одной или нескольких труппах, а в каждой труппе должен быть хотя бы один актер. Ассоциация отображает, что каждой труппой управляет только один актер — художественный руководитель, а некоторые актеры не являются руководителями.

диаграмма классов информационной системы театра

- Ассоциация между классами Спектакль и Актер фиксирует, что в спектакле должен быть занят хотя бы один актер, впрочем, актер может играть в любом количестве спектаклей (или вообще может ничего не играть).
- Между классами Спектакль и Зритель тоже определена ассоциация. Она поясняет, что зритель может смотреть любое число спектаклей, а на каждом спектакле может быть любое число зрителей.
- И наконец, на диаграмме отображены два отношения наследования, утверждающие, что и в зрителях, и в актерах есть человеческое начало.

Контрольные вопросы

- Поясните назначение статических моделей объектно-ориентированных программных систем.
- Что является основным средством для представления статических моделей?
- Как используются статические модели?
- Какие секции входят в графическое обозначение класса?
- Какие секции класса можно не показывать?
- Какие имеются разновидности области действия свойства (операции)?
- Поясните общий синтаксис представления свойства.
- Какие уровни видимости вы знаете? Их смысл?
- Какие характеристики свойств вам известны?
- Поясните общий синтаксис представления операции.
- Какой вид имеет форма представления параметра операции?
- Какие характеристики операций вам известны?
- Что означают три точки в списке свойств (операций)?
- Как организуется группировка свойств (операций)?
- Как ограничить количество экземпляров класса?

Контрольные вопросы

- Перечислите известные вам «украшения» отношения ассоциации.
- Может ли статическая модель программной системы не иметь отношений ассоциации?
- Какой смысл имеет квалификатор? К чему он относится?
- Какие отношения могут иметь пометки видимости и что эти пометки обозначают?
- Какой смысл имеет класс-ассоциация?
- Чем отличается агрегация от композиции? Разновидностями какого отношения (в UML) они являются?
- Что обозначает в UML простая зависимость?
- Какой смысл имеет отношение обобщения?
- Какие недостатки у множественного наследования?
- Перечислите недостатки ромбовидной решетки наследования.
- В чем смысл отношения реализации?
- Что обозначает мощность «многие-ко-многим» и в каких отношениях она применяется?
- Что такое абстрактный класс (операция) и как он (она) отображается?
- Как запретить полиморфизм операции?
- Как обозначить корневой класс?