1. **Диаграммы классов** показывают набор **классов**, интерфейсов, а также их связи. **Диаграммы** этого вида чаще всего используются для моделирования объектно-ориентированных систем. Они предназначены для статического представления системы.

2. Каким образом классы в UML.

Графически **класс** изображается в виде прямоугольника, разделенного на 3 блока горизонтальными линиями: имя **класса** атрибуты (свойства) **класса** операции (методы) **класса**.

.3. Расскажите об атрибутах класса.

Атрибут - это именованное свойство класса, включающее описание множества значений, которые могут принимать экземпляры этого свойства. Класс может иметь любое число атрибутов или не иметь их вовсе. Атрибут представляет некоторое свойство моделируемой сущности, общее для всех объектов данного класса. В каждый момент времени любой атрибут объекта, принадлежащего данному классу, обладает вполне определенным значением. Атрибуты представлены в разделе, который расположен под именем класса; при этом указываются только их имена.

4. Расскажите об операциях.

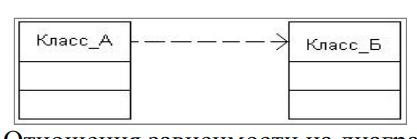
Операцией называется реализация услуги, которую можно запросить у любого объекта класса для воздействия на поведение. Иными словами, операция – это абстракция того, что позволено делать с объектом. У всех объектов класса имеется общий набор операций. Класс может содержать любое число операций или не содержать их вовсе. Часто (хотя не всегда) обращение к операции объекта изменяет его состояние или его данные. Операции класса изображаются в разделе, расположенном ниже раздела с атрибутами.

.5. Опишите основные отношения между классами.

- отношение зависимости (dependency relationship);

Отношение зависимости в общем случае указывает некоторое семантическое отношение между двумя элементами модели или двумя множествами таких элементов, которое не является отношением ассоциации, обобщения или реализации. Отношение зависимости используется в такой ситуации, когда некоторое изменение одного элемента модели может потребовать изменения другого зависимого от него элемента модели.

- "access" – служит для обозначения доступности открытых атрибутов и операций класса-источника для классов-клиентов; - "bind" – класс-клиент может использовать некоторый шаблон для своей последующей параметризации; - "derive" – атрибуты класса-клиента могут быть вычислены по атрибутам класса-источника; - "import" – открытые атрибуты и операции класса-источника становятся частью класса-клиента, как если бы они были объявлены непосредственно в нем; - "refine" – указывает, что класс-клиент служит уточнением классаисточника в силу причин исторического характера, когда появляется дополнительная информация в ходе работы над проектом.



Класс а зависит от класса б

- отношение ассоциации (association relationship);

Ассоциация - это отношение, при котором объекты одного типа неким образом связаны с объектами другого типа. Например, объект одного типа содержит или использует объект другого типа.

Отношение ассоциации соответствует наличию некоторой связи между классами. Данное отношение обозначается сплошной линией с дополнительными специальными символами, которые характеризуют отдельные свойства конкретной ассоциации.



Сотрудник работает в компании

- отношение обобщения (generalization relationship).

Применительно к диаграмме классов данное отношение описывает иерархическое строение классов и наследование их свойств и поведения. При этом предполагается, что класс-потомок обладает всеми свойствами и поведением класса-предка, а также имеет свои собственные свойства и поведение, которые отсутствуют у класса-предка.



.6. Дайте определение системы и расскажите о ее свойствах.

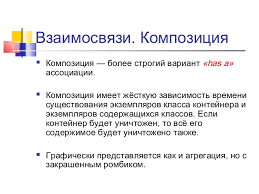
Агрегация

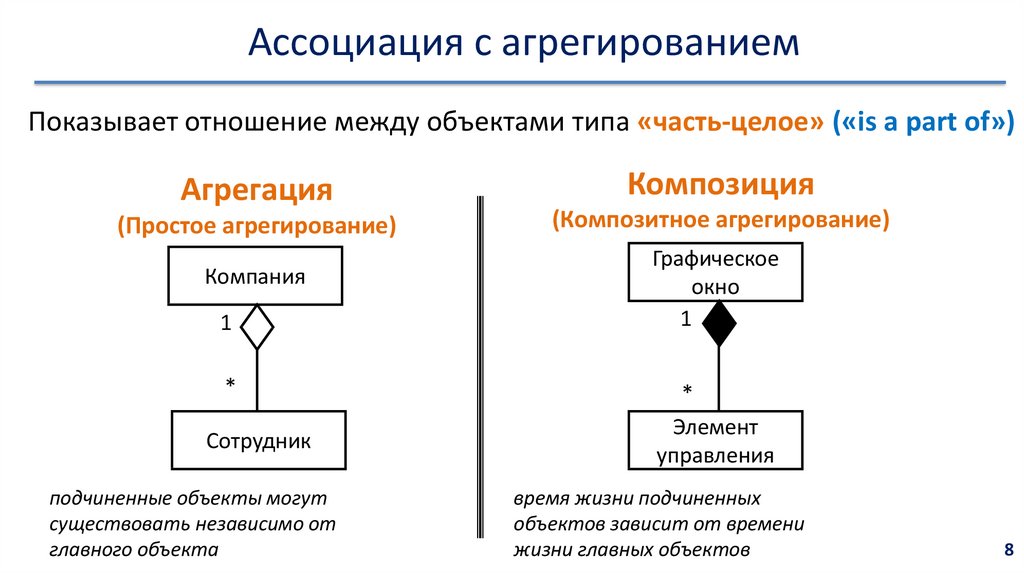
Агрегация является подтипом отношения ассоциации в UML. Агрегация и состав являются типами ассоциативных отношений в UML. Отношения агрегации можно описать простыми словами как «объект одного класса может владеть или получать доступ к объектам другого класса».

В отношении агрегации зависимый объект остается в области отношения, даже если исходный объект уничтожен.

Композиция

Композиция является двусторонним отношением, которое также называется бинарным отношением.





+полиморфизм

**Полиморфизм** - это способность объекта использовать методы производного класса, который не существует на момент создания базового.

Используем одно и то же разными способами.