1. **Какие отношения возможны между классами?**

* обобщение/специализация (generalization/specialization)

кошки — это животные

* + Наследование
  + Стрелка всегда указывает на базовый (родительский) класс.
  + Наследования - сильный вид отношения, поэтому к нему следует прибегать с осторожностью рассматривая в качестве альтернативы агрегацию.
* целое/часть (whole/part) двигатель - часть автомобиля  
  + Агрегация: агрегирующий объект (в примере list) может не включать в себя экземпляры агрегируемого (в примере atom)
  + Композиция: один объект (в примере Rectangle) не может существовать без своих составляющих (в примере Point)
  + Композиция - более строгий вариант агрегации.
  + Композиция имеет жёсткую зависимость времени существования экземпляров класса контейнера и экземпляров содержащихся классов. Если контейнер будет уничтожен, то всё его содержимое будет также уничтожено.
  + Агрегация: профессора - факультеты, профессора остаются жить после разрушения факультета
  + Композиция: университет - факультеты, факультеты без университета погибают.
* ассоциация (семантическая зависимость)  
  художник - кисть
* Ассоциация - самый слабый вид отношения из перечислены.
* Как правило при ассоциации один класс так или иначе использует другой, но не владеет им.
* Например, один класс может хранить ссылку на другой, однако при удалении объекта первого класса, второй продолжит существовать

1. **Какие отношения возможны между объектами?**
2. **Что такое мощность отношения?**

* Мощность отношения (мультипликатор) означает число связей между каждым экземпляром класса (объектом) в начале линии с экземпляром класса в её конце.
* Мощность:
  + 0..1 Ноль или один экземпляр

кошка имеет или не имеет хозяина

* + 1 Обязательно один экземпляр  
    у кошки одна мать
  + 0..\* или \* Ноль или более экземпляров  
    у кошки могут быть, а может и не быть котят
  + 1..\* Один или более экземпляров  
    у кошки есть хотя бы одно место, где она спит

1. **Как в C++ может быть реализовано отношение ассоциация (связь)?**

* Поле одного класса может быть указателем на другой класс

1. **Чем отличается агрегация от композиции?**

* Агрегация: профессора - факультеты, профессора остаются жить после разрушения факультета
* Композиция: университет - факультеты, факультеты без университета погибают.

1. **Как в C++ может быть реализовано отношения агрегации и композиции?**

* Агрегация

class Part{

// ...

};

class Big{

Part \*filed1;

}

* Класс Big может хранить в себе указатель на Part, но экземпляр класса Big можно создать и не создавая экземпляра Part;
* Если того требует мощность отношения в Big можно включить несколько экземпляров класса Part, например использовав массив:

vector<Part\*> filed1;

class Doctor{

// ...

};

class Patient{

Doctor \*filed1;

}

* Класс Patient знает про ассоциированный с ним объект типа Doctor.
* Может менять ссылку на Doctor, но не может удалять экземпляр класса Doctor. Ассоциация не потеряет своего смысла, если в классе Doctor добавить указатель (или массив указателей) на Patient.
* Синтаксически, эта форма ассоциации, может быть похожа на агрегацию.
* Однако объекты здесь более независимы. В зависимости от своего характера ассоциация, может быть представлена и иначе.
* Композиция

class Part{

// ...

};

class Big{

Part filed1;

}

* Big включает в себя Part. Причем при создании экземпляра класса Big будет обязательно создан экземпляр класса Part;
* Если того требует мощность отношения в Big можно включить несколько экземпляров класса Part, например использовав массив.

1. **Когда следует использовать наследование, а когда агрегацию (композицию)?**

* Наследования - сильный вид отношения, поэтому к нему следует прибегать с осторожностью рассматривая в качестве альтернативы агрегацию.