

## **1. Перечислите и прокомментируйте основные варианты отношений между классами.**

а) композиция: один из классов использует и контролирует другой (один экземпляр первого владеет одним экземпляром второго)

б) агрегация: один из классов использует другой, но не контролирует его, поскольку один экземпляр второго класса, возможно, управляется несколькими экземплярами первого класса.

в) ассоциация: классы используют друг друга, не являются «частью-целым», экземпляры имеют связь.

г) зависимость: происходит периодическое использование одного класса другим без связи.

## **2. Какие существуют разновидности наследования и для чего они предназначены?**

а) интерфейса (открытое): derived класс сохраняет все спецификаторы доступа класса base и добавляет что-то свое, тем самым представляя из себя «разновидность» base.

б) реализации (закрытое): derived класс переводит все унаследованное в private, а значит, реализован посредством интерфейса base класса, за исключением собственного нового интерфейса.

## **3. Что необходимо для корректного функционирования механизма виртуальных функций?**

1) Использование указателей с динамическим типом.

2) Прописывание virtual перед перегруженной функцией как минимум в base, а лучше везде.

3) Полное совпадение сигнатур функций (проверяем через override).

## **4. Какую проблему решают виртуальные базовые классы при множественном наследовании?**

Виртуальные базовые классы решают такую проблему, когда один наследник нескольких наследников базового класса содержит в себе

базовый класс несколько раз (через каждого из родителей). С добавлением виртуальности базовый класс не дублируется.

## **5. Перечислите и прокомментируйте основные категории паттернов проектирования.**

1) Порождающие — решают проблему создания экземпляров классов путем создания специальных классов-фабрик, отвечающих за это. Важно при переписывании кода, тогда достаточно изменить лишь фрагмент из фабрики.

2) Структурные — отвечают за построение иерархий, удобных для поддержки и использования.

3) Поведения — рассматривают вопросы распределения обязанностей между объектами, последовательность выполнения ими некоторых действий.