

## Ответы на контрольные вопросы:

### 1 вопрос: Перечислите и прокомментируйте основные варианты отношений между классами

Рассмотрим такие классы, как цветок, роза, лепесток и божья коровка. Можно заметить, что хоть все классы разные, между некоторыми из них определенно есть взаимосвязь. Опишем основные типы отношений между классами на этих примерах.

Есть два основных типа отношений между классами:

→ "Обобщение/специализация" (общее и частное), известное как *is-a*. Роза - это цветок, то есть класс роза - это подкласс более общего класса цветы. При таком отношении один класс (частное) расширяет возможности другого класса (общее). Расширение возможностей класса осуществляется благодаря использованию наследования.

→ Отношение, при котором существует взаимосвязь между классами:

а) Отношение *has-a*, при котором класс содержит другой класс (роза имеет лепестки). В этом случае в классе объявляется один или несколько экземпляров другого класса. При данном отношении возможны два случая взаимодействия. Первый случай, это когда объект (экземпляр), который объявлен в классе, не является составной частью класса (агрегация) и его использование не влияет на функциональную работу класса. Второй случай, когда объект, который объявлен в классе, является составной частью этого класса (композиция).

б) Отношение типа *uses*. (Роза использует божьих коровок для опыления). В этом случае класс содержит программный код другого вложенного класса, к которому он имеет доступ.

### 2 вопрос: Какие существуют разновидности наследования и для чего они предназначены?

*Важно:* при наследовании спецификатор доступа члена родительского класса может быть изменен в дочернем классе (в зависимости от типа наследования).

Есть три типа наследования:

→ **public** - публичное: публичные и защищенные данные наследуются без изменения уровня доступа к ним, приватные данные недоступны (то есть всё остается без изменений). Такое наследование является самым распространенным.

→ **protected** - защищенное: все унаследованные данные становятся защищенными. Этот тип используется очень редко.

→ **private** - приватное: все унаследованные данные становятся приватными (приватные по-прежнему недоступны). Закрытое наследование может быть полезно, когда дочерний класс не имеет очевидной связи с родительским классом, но использует его в своей реализации. В таком случае мы не хотим, чтобы открытый интерфейс родительского класса был доступен через объекты дочернего класса (то есть защищаем данные родительского класса от изменений).

### 3 вопрос: Что необходимо для корректного функционирования механизма виртуальных функций?

Когда в базовом классе объявляется хотя бы одна виртуальная функция, то для всех полиморфных классов создается таблица виртуальных функций (virtual function table).

Таблица виртуальных функций – это одномерный массив указателей на функции. Количество элементов в массиве равно количеству виртуальных функций в классе. Виртуальные функции нужны для поддержки полиморфизма, который позволяет использовать одинаковый синтаксис для разных классов.

→ Так как виртуальные функции нужны для переопределения, то необходимо, чтобы в дочерних классах это переопределение как минимум было (и сигнатура виртуального метода дочернего класса должна полностью соответствовать сигнатуре виртуального метода родительского класса. Если у дочернего метода будет другой тип параметров, нежели у родительского, то вызываться этот метод не будет).

→ Еще для корректного функционирования необходимо правильное исходное определение (классовое)

объекта, к которому хотелось бы применить виртуальную функцию.

→ Типы возврата виртуальной функции и её переопределений должны совпадать (пункт один про сигнатуру).

→ Нельзя вызывать виртуальные функции в теле конструкторов и деструкторов.

→ Если в классе есть хотя бы одна виртуальная функция, то деструктор тоже должен быть виртуальным, поэтому его нужно объявить явно.

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

Проблему ромба: пусть классы В и С наследуют класс А, а класс D наследует В и С.

Тогда если три класса А, В и С будут определять какой-нибудь одинаковый метод, который потом будет вызываться классом D, то будет непонятно - какой из методов (класса А, класса В или класса С) должен быть вызван - решается виртуальными функциями.

Если мы создадим объект класса D, то на выходе получим сразу две копии класса D - одну от В и одну от С. В случаях, когда копия нужна всего одна, как раз используют родительские виртуальные классы, в которых объект является общим для всех дочерних классов.

#### **5 вопрос: Перечислите и прокомментируйте основные категории паттернов проектирования**

(for me) *Паттернами проектирования (Design Patterns) называют решения часто встречающихся проблем в области разработки программного обеспечения. Паттерны проектирования не являются готовыми решениями, которые можно трансформировать непосредственно в код, а представляют общее описание решения проблемы, которое можно использовать в различных ситуациях.* (end for me)

Есть несколько основных категорий:

→ Порождающие паттерны - предназначены для создания новых объектов в системе.

→ Структурные паттерны - решают задачи компоновки системы на основе классов и объектов.

→ Паттерны поведения - предназначены для распределения обязанностей между объектами в системе.