

Класс проектирования – это наиболее приближенная к реальности абстракция класса или похожей конструкции реализации системы. Наиболее близкой к реальности по следующим причинам: 1. Язык, используемый для описания класса проектирования, тот же, что и язык программирования для реализации. Соответственно, операции, параметры, атрибуты, типы и другие подробности определяются с использованием синтаксиса выбранного языка программирования.

2. Часто задается видимость атрибутов и операций класса проектирования. Например, в языке C++ для этого обычно используются ключевые слова public, protected и private.

3. Отношения с другими классами, в которых участвует данный класс проектирования, часто получают явное выражение при реализации этого класса. Например, обобщение во всех стереотипах обобщения имеет семантику, которая соответствует обобщению в языке программирования. Таким образом, обобщения и агрегации часто отображаются на соответствующие переменные (атрибуты) реализации, соответствующие ссылкам на объекты.

4. Методы класса проектирования прямо отображаются на соответствующие методы классов реализации. Методы, определяемые в ходе проектирования, часто определяются на естественном языке или на псевдокоде и могут быть в таком виде использованы в аннотации к реализации этих методов. Это одна из немногих серьезных абстракций, переходящих из проектирования в реализацию, и непосредственная причина нашей рекомендации задействовать для выполнения проектирования и реализации класса одного и того же разработчика.

5. Класс проектирования может переложить обработку некоторых требований на последующую реализацию, передав ей требования к реализации класса. В результате становится возможным отложить принятие решений, которые невозможно сделать на

основе модели проектирования, например, касающихся вопросов кодирования класса.

6. Класс проектирования часто задается стереотипом, который напрямую отображается в конструкцию соответствующего языка программирования.

7. Класс проектирования может быть реализован — и реализуется — в виде интерфейса, если это понятие существует в выбранном языке программирования. Например, класс проектирования, представляющий класс языка Java, может быть реализован в виде интерфейса.

8. Класс проектирования может быть активным. Это означает, что объекты класса будут использовать свою собственную нить управления, работая параллельно с другими активными объектами. Однако обычно классы проектирования неактивны. Это означает, что все их объекты «запускаются» в едином адресном пространстве под управлением другого активного объекта. Детали семантики этого процесса зависят от языка программирования и используемых в нем технологий параллельной или распределенной обработки. Отметим важность разницы между активными классами и неактивными.

Активные классы могут, следовательно, в качестве альтернативы, содержаться не только в модели проектирования, но и в их собственной модели процесса. Такое характерно для случая множества активных классов, объекты которых тесно интегрированы, - например, в системах реального времени.