**8 UML. Основы структурного моделирования**

Выскажите общее суждение о языке моделирования программных систем – UML. Перечислите диаграммы языка UML, используемые для структурного моделирования и опишите их. Охарактеризуйте диаграмму классов, приведите примеры.

**1. Выскажите общее суждение о языке моделирования программных систем – UML.**

Унифицированный язык моделирования (UML) является стандартным инструментом для создания "чертежей" программного обеспечения. С помощью UML можно визуализировать, специфицировать, конструировать и документировать артефакты программных систем.

UML пригоден для моделирования любых систем: от информационных систем масштаба предприятия до распределенных Web-приложений и даже встроенных систем реального времени. UML – это язык, позволяющий рассмотреть систему со всех точек зрения, имеющих отношение к ее разработке и последующему развертыванию.

Сфера применения UML не ограничивается моделированием программного обеспечения. Его выразительность позволяет моделировать, скажем, документооборот в юридических системах, структуру и функционирование системы обслуживания пациентов в больницах, осуществлять проектирование аппаратных средств.

**2. Перечислите диаграммы языка UML, используемые для структурного моделирования и опишите их.**

В UML существует четыре структурных диаграммы для визуализации, специфицирования, конструирования и документирования статических аспектов системы, составляющих ее относительно прочный "костяк". Подобно тому как статические аспекты дома показывают, что и каким образом будет размещено в здании (стены, двери, окна, трубы и т.д.), так и статические аспекты программных систем отражают наличие и расположение классов, интерфейсов, коопераций, компонентов, узлов и других сущностей.

Названия структурных диаграмм UML соответствуют названиям основных групп сущностей, используемых при моделировании системы:

* диаграммы классов - классам, интерфейсам и кооперациям;
* диаграммы объектов - объектам;
* диаграммы компонентов - компонентам;
* диаграммы развертывания - узлам.

На *диаграмме классов* изображают множество классов, интерфейсов, коопераций и их отношений. Это самый распространенный тип диаграмм, применяемый при моделировании объектно-ориентированных систем; он используется для иллюстрации статического вида системы с точки зрения проектирования. Диаграммы, на которых показаны активные классы, применяются для работы со статическим видом системы с точки зрения процессов.

На *диаграмме объектов* показывают множество объектов и отношения между ними. Такие изображения используются для иллюстрации структуры данных, то есть статических "мгновенных снимков" экземпляров тех сущностей, которые представлены на диаграмме классов. Диаграммы объектов относятся к статическому виду системы с точки зрения процессов, заостряют внимание на реальных или модельных прецедентах.

На *диаграммах компонентов* показаны множества компонентов и отношения между ними. С их помощью иллюстрируют статический вид системы с точки зрения реализации. Диаграммы компонентов соотносятся с диаграммами классов, так как обычно компонент отображается на один или несколько классов, интерфейсов или коопераций.

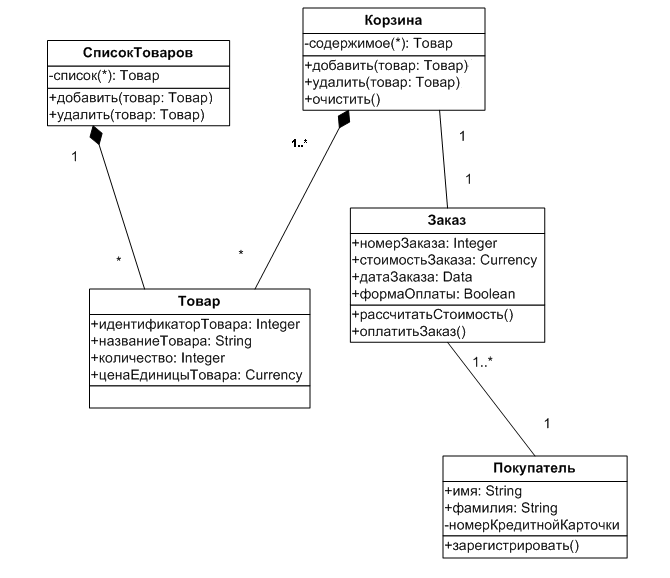
На *диаграммах развертывания* представлены узлы и отношения между ними. С помощью таких изображений иллюстрируют статический вид системы с точки зрения развертывания.

Они соотносятся с диаграммами компонентов, так как узел обычно содержит один или несколько компонентов.

**3. Охарактеризуйте диаграмму классов и диаграмму объектов, приведите примеры.**

Диаграммы классов – это наиболее часто используемый тип диаграмм, которые создаются при моделировании объектнооориентированных систем. Показывают набор классов, интерфейсов и коопераций, а также их связи. Диаграммы классов важны не только для визуализации, специфицирования и документирования структурных моделей, но также для конструирования исполняемых систем посредством прямого и обратного проектирования.На диаграммах классов обычно представлены следующие элементы: классы, интерфейсы, зависимости, обобщения и ассоциации. Рассмотрим базовые составляющие:

|  |  |
| --- | --- |
| *Класс* –это описание совокупности объектов с общими атрибутами, операциями, отношениями и семантикой. *Атрибут* - это именованное свойство класса, включающее описание множества значений, которые могут принимать экземпляры этого свойства. Класс может иметь любое число атрибутов. *Операцией* - это абстракция того, что позволено делать с объектом. Класс может содержать любое число операций или не содержать их вовсе. |  |
| *Отношение ассоциации* отображаются в виде линии, соединяющей взаимодействующие классы, и могут показывать роль и количество объектов с каждой стороны. |  |
| *Отношение агрегации* являются специальным типом ассоциаций, в котором два участвующих класса не равнозначны по статусу, но создают взаимодействие типа «часть-целое». | Леоненков |
| *Отношение композиции* частный случай агрегации. Части не могут выступать в отрыве от целого, т.е. с уничтожением целого уничтожаются составные части. | Леоненков |
| *Отношение обобщения.* Наследование является основной концепцией объектно-ориентированного программирования, в которой класс «получает» все атрибуты и методы наследуемого класса и может модифицировать некоторые из них, имея возможность добавлять собственные атрибуты и методы. В UML ассоциация *обобщения* между двумя классами помещает их в иерархию, представляющую концепцию наследования производного класса от базового. | Леоненков |
| *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. | Леоненков |

Пример: диаграмма классов системы

Продажи товаров в интернет-магазине.