**Практическое** **занятие** **№12**

**Группа**: П-16

**Учащийся**: Шелест Анатолий

**Дата**: 21.11.2022

**Тема**: Унифицированный язык моделирования UML. Концептуальная модель

**1. Цель работы:** Изучить состав концептуальной модели .UML, строительные блоки, правила языка и общие механизмы.

**2. Задание:** Изучить теоретический материал для создания концептуальной модели.

**Ход работы**

**Задание 1**

Для понимания UML необходимо усвоить его концептуальную модель, которая включает в себя три составные части: основные строительные блоки языка, правила их сочетания и некоторые общие для всего языка механизмы. Усвоив эти элементы, вы сумеете читать модели на UML и самостоятельно создавать их - вначале, конечно, не очень сложные. По мере приобретения опыта в работе с языком вы научитесь пользоваться и более развитыми его возможностями.

Сущности - это абстракции, являющиеся основными элементами модели. Отношения связывают различные сущности; диаграммы группируют представляющие интерес совокупности сущностей.

В UML имеется четыре типа сущностей:

\* структурные;

\* поведенческие;

\* группирующие;

\* аннотационные.

Сущности являются основными объектно-ориентированными блоками языка. С их помощью можно создавать корректные модели.

Структурные сущности - это имена существительные в моделях на языке UML. Как правило, они представляют собой статические части модели, соответствующие концептуальным или физическим элементам системы.

Строительные блоки UML нельзя произвольно объединять друг с другом. Как и любой другой язык, UML характеризуется набором правил, определяющих, как должна выглядеть хорошо оформленная модель, то есть семантически самосогласованная и находящаяся в гармонии со всеми моделями, которые с нею связаны.

В языке UML имеются семантические правила, позволяющие корректно и однозначно определять:

\* имена, которые можно давать сущностям, отношениям и диаграммам;

\* область действия (контекст, в котором имя имеет некоторое значение);

\* видимость (когда имена видимы и могут использоваться другими элементами);

\* целостность (как элементы должны правильно и согласованно соотноситься друг с другом);

\* выполнение (что значит выполнить или имитировать некоторую динамическую модель).

**Общие механизмы языка UML**

Работу с этим языком существенно облегчает последовательное использование общих механизмов, перечисленных ниже:

спецификации (Specifications);

дополнения (Adornments);

принятые деления (Common divisions);

механизмы расширения (Extensibility mechanisms).

UML - это не просто графический язык. За каждой частью его системы графической нотации стоит спецификация, содержащая текстовое представление синтаксиса и семантики соответствующего строительного блока. С помощью графической нотации UML вы визуализируете систему, с помощью спецификаций UML - описываете ее детали. Таким образом, допустимо строить модель инкрементно, то есть пошаговым образом - сначала нарисовать диаграмму, а потом добавить семантику в спецификацию модели, или наоборот - начать со спецификации (возможно, применив обратное проектирование к существующей системе), а потом на ее основе создавать диаграммы.

Спецификации UML создают семантический задний план, который полностью включает в себя составные части всех моделей системы, согласованные между собой. Таким образом, диаграммы UML можно считать визуальными проекциями на этот задний план, при этом каждая из них раскрывает один из значимых аспектов системы.

**Механизмы расширения.**

UML - это стандартный язык для разработки "чертежей" программного обеспечения, но ни один замкнутый язык не в состоянии охватить нюансы всех возможных моделей в различных предметных областях. Поэтому UML является открытым языком, то есть допускает контролируемые расширения. Механизмы расширения UML включают:

\* стереотипы;

\* помеченные значения;

\* ограничения.

Стереотип (Stereotype) расширяет словарь UML, позволяя на основе существующих блоков языка создавать новые, специфичные для решения конкретной проблемы. Например, работая с такими языками программирования, как Java или C++, часто приходится моделировать исключения (Exceptions) - они являются обыкновенными классами, хотя и рассматриваются особым образом. Обычно требуется, чтобы исключения можно было возбуждать и перехватывать, и ничего больше. Если пометить исключения соответствующим стереотипом, то с ними можно будет обращаться как с обычными строительными блоками языка.

Помеченное значение (Tagged value) расширяет свойства строительных блоков UML, позволяя включать новую информацию в спецификацию элемента. Скажем, если вы работаете над "коробочным" продуктом и выпускаете много его версий, то зачастую необходимо отслеживать версию и автора какой-нибудь важной абстракции. Ни версия, ни автор не являются первичными концепциями UML, но их можно добавить к любому блоку, такому, например, как класс, задавая для него новые помеченные значения.

**Задание 2**

Словарь языка UML включает три вида строительных блоков:

1) сущности;

2) отношения;

3) диаграммы.

Сущности - это абстракции, являющиеся основными элементами модели. Отношения связывают различные сущности; диаграммы группируют представляющие интерес совокупности сущностей.

В UML имеется четыре типа сущностей:

1) структурные;

2) поведенческие;

3) группирующие;

4) аннотационные.

Сущности являются основными объектно-ориентированными блоками языка.. С их помощью можно создавать корректные модели.

Структурные сущности - это имена существительные в моделях на языке UML. Как правило, они представляют собой статические части модели, соответствующие концептуальным или физическим элементам системы. Существует семь разновидностей структурных сущностей.

Класс (Class) - это описание совокупности объектов с общими атрибутами, операциями, отношениями и семантикой. Класс реализует один или несколько интерфейсов.

Интерфейс (Interface) - это совокупность операций, которые определяют сервис (набор услуг), предоставляемый классом или компонентом. Таким образом, интерфейс описывает видимое извне поведение элемента. Интерфейс может представлять поведение класса или компонента полностью или частично; он определяет только спецификации операций (сигнатуры), но никогда - их реализации. Графически интерфейс изображается в виде круга, под которым пишется его имя. Интерфейс редко существует сам по себе - обычно он присоединяется к реализующему его классу или компоненту.

Кооперация (Collaboration) определяет взаимодействие; она представляет собой совокупность ролей и других элементов, которые, работая совместно, производят некоторый кооперативный эффект, не сводящийся к простой сумме слагаемых. Кооперация, следовательно, имеет как структурный, так и поведенческий аспект. Один и тот же класс может принимать участие в нескольких кооперациях; таким образом, они являются реализацией образцов поведения, формирующих систему.

Прецедент (Use case) - это описание последовательности выполняемых системой действий, которая производит наблюдаемый результат,значимый для какого-то определенного актера (Actor). Прецедент применяется для структурирования поведенческих сущностей модели. Прецеденты реализуются посредством кооперации.

Три другие сущности - активные классы, компоненты и узлы - подобны классам: они описывают совокупности объектов с общими атрибутами, операциями, отношениями и семантикой. Тем не менее они в достаточной степени отличаются друг от друга и от классов и, учитывая их важность при моделировании определенных аспектов объектно-ориентированных систем, заслуживают специального рассмотрения.

Активным классом (Active class) называется класс, объекты которого вовлечены в один или несколько процессов, или нитей (Threads), и поэтому могут инициировать управляющее воздействие. Активный класс во всем подобен обычному классу, за исключением того, что его объекты представляют собой элементы, деятельность которых осуществляется одновременно с деятельностью других элементов.

Два оставшихся элемента - компоненты и узлы - также имеют свои особенности. Они соответствуют физическим сущностям системы, в то время как пять предыдущих - концептуальным и логическим сущностям.

Компонент (Component) - это физическая заменяемая часть системы, которая соответствует некоторому набору интерфейсов и обеспечивает его реализацию. В системе можно встретить различные виды устанавливаемых компонентов, такие как СОМ+ или Java Beans, а также компоненты, являющиеся артефактами процесса разработки, например файлы исходного кода. Компонент, как правило, представляет собой физическую

упаковку логических элементов, таких как классы, интерфейсы и кооперации.

Узел (Node) - это элемент реальной (физической) системы, который существует во время функционирования программного комплекса и представляет собой вычислительный ресурс, обычно обладающий как минимум некоторым объемом памяти, а часто еще и способностью обработки (см. главу 26). Совокупность компонентов может размещаться в узле, а также мигрировать с одного узла на другой.

**Задание 3**

1. Дайте определение «UML»

Ответ: UML – унифицированный язык визуального моделирования, который является языком широкого профиля; это открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования в основном программных систем.

2. Перечислите типы отношений

Ответ: В языке UML определены четыре типа отношений:

\* зависимость;

\* ассоциация;

\* обобщение;

\* реализация.

3. Перечислите механизмы расширения UML

Ответ: Механизмами расширения в UML являются:

1) Ограничения - позволяют определять новые или изменять существующие правила

2) Помеченные значения - позволяют включать новую информацию в спецификацию элемента

3) Стереотипы - расширяют словарь UML, позволяя на основе существующих блоков языка создавать новые, специфичные для решения конкретной проблемы

4. Где используется язык UML?

Ответ: Язык UML предназначен прежде всего для разработки программных систем. Его использование особенно эффективно в следующих областях:

· информационные системы масштаба предприятия;

· банковские и финансовые услуги;

· телекоммуникации;

· транспорт;

· оборонная промышленность, авиация и космонавтика;

· розничная торговля;

· медицинская электроника;

· наука;

· распределенные Web-системы.

5. Дайте определение «Сущность»

Ответ: Сущности - это абстракции, являющиеся основными элементами модели. Сущности являются основными объектно-ориентированными блоками языка.. С их помощью можно создавать корректные модели.

6. Перечислите строительные блоки UML.

Ответ: Словарь языка UML включает три вида строительных блоков:

\* сущности;

\* отношения;

\* диаграммы.

7. Перечислите виды общих механизмов UML.

Ответ: Работу с этим языком существенно облегчает последовательное использование общих механизмов, перечисленных ниже:

1) спецификации (Specifications);

2) дополнения (Adornments);

3) принятые деления (Common divisions);

4) механизмы расширения (Extensibility mechanisms).

8. Для чего используются механизмы расширения языка UML.

Ответ: Механизмы расширения языка позволяют модифицировать UML в соответствии с потребностями вашего проекта. Кроме того, они дают возможность адаптировать UML к новым технологиям разработки программного обеспечения, например к вероятному появлению более мощных языков распределенного программирования. С помощью механизмов расширения можно создавать новые строительные блоки, модифицировать существующие и даже изменять их семантику.