**Занятие № 12**

**Номер учебной группы:** П-16.

**Фамилия, инициалы учащегося:** Новалихина С. К.

**Дата выполнения работы:** 21.11.2022.

**Тема работы:** «Унифицированный язык моделирования UML. Концептуальная модель»

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

**Ход работы**

**Задание 1**

Изучил теоретический материал по теме «Концептуальная модель UML».

Для понимания UML необходимо усвоить его концептуальную модель, которая включает в себя три составные части:

– основные строительные блоки языка;

– правила их сочетания;

– некоторые общие для всего языка механизмы.

Словарь языка UML включает три вида строительных блоков:

– сущности;

– отношения;

– диаграммы.

***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;

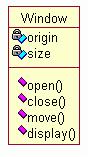
– группирующие;

– 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. Существуетнесколько разновидностей структурных сущностей.

**Класс** (Class) – это описание совокупности объектов с общими атрибутами, операциями, отношениями и семантикой. Класс реализует один или несколько интерфейсов. **Графически класс изображается в виде прямоугольника**, в котором обычно записаны его имя, атрибуты и операции, как показано на рисунке.



**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, как показано на рисунке. Интерфейс редко существует сам по себе – обычно он присоединяется к реализующему его классу или компоненту.



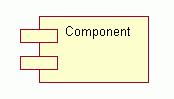
**Кооперация** (Collaboration) определяет взаимодействие; она представляет собой совокупность ролей и других элементов, которые, работая совместно, производят некоторый кооперативный эффект, не сводящийся к простой сумме слагаемых. Кооперация, следовательно, имеет как структурный, так и поведенческий аспект. Один и тот же класс может принимать участие в нескольких кооперациях; таким образом, они **являются реализацией образцов поведения**, формирующих систему. **Графически кооперация изображается в виде эллипса**, ограниченного пунктирной линией, в который обычно заключено только имя, как показано на рисунке.



**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. Графически **прецедент изображается в виде ограниченного непрерывной линией эллипса**, обычно содержащего только его имя, как показано на рисунке.



**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. **Графически компонент изображается в виде прямоугольника с вкладками**, содержащего обычно только имя, как показано на рисунке. Компонентподобен классу: он описывает совокупность объектов с общими атрибутами, операциями, отношениями и семантикой.



Этибазовые элементы – **классы, интерфейсы, кооперации, прецеденты и компоненты** – являются основными структурными сущностями, которые могут быть включены в модель UML Существуют также разновидности этих сущностей: **актеры, сигналы, утилиты** (виды классов), **процессы** и **нити** (виды активных классов), **приложения**, **документы**, **файлы**, **библиотеки**, **страницы** и **таблицы** (виды компонентов).

**Поведенческие сущности** (Behavioral things) являются динамическими составляющими модели UML. Это глаголы языка: они описывают поведение модели во времени и пространстве. Существует всего два основных типа поведенческих сущностей.

*Взаимодействие* (Interaction) – это поведение, суть которого заключается в обмене **сообщениями** (Messages) между объектами в рамках конкретного контекста для достижения определенной цели. С помощью взаимодействия можно описать как отдельную операцию, так и поведение совокупности объектов. Взаимодействие предполагает ряд других элементов, таких как сообщения, последовательности действий (поведение, инициированное сообщением) и связи (между объектами). **Графически сообщения изображаются в виде стрелки**, над которой почти всегда пишется имя соответствующей операции, как показано на рисунке.

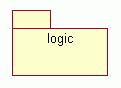
http://www.maksakov-sa.ru/files/12.JPG

*Автомат* (State machine) – это алгоритм поведения, определяющий **последовательность состояний**, через которые объект или взаимодействие проходят на протяжении своего жизненного цикла в ответ на различные события, а также реакции на эти события. С помощью автомата можно **описать поведение отдельного класса или кооперации классов**. С автоматом связан ряд других элементов: состояния, переходы (из одного состояния в другое), события (сущности, инициирующие переходы) и виды действий (реакция на переход). **Графически состояние изображается в виде прямоугольника с закругленными углами**, содержащего имя и, возможно, подсостояния.

http://www.maksakov-sa.ru/files/13.JPG

Эти два элемента – **взаимодействия и автоматы** – являются основными поведенческими сущностями, входящими в модель UML. Семантически они часто бывают связаны с различными структурными элементами, в первую очередь – классами, кооперациями и объектами.

**Группирующие сущности** являются организующими частями модели UML. Это блоки, на которые можно разложить модель. Есть только одна первичная группирующая сущность, а именно пакет.



**Пакеты** (Packages) представляют собой универсальный механизм организации элементов в группы. В пакет можно поместить структурные, поведенческие и даже другие группирующие сущности. В отличие от компонентов, существующих во время работы программы, пакеты носят чисто концептуальный характер, то есть существуют только во время разработки. **Изображается пакет в виде папки с закладкой**, содержащей, как правило, только имя и иногда – содержимое.

**Пакеты** – это основные группирующие сущности, с помощью которых можно организовать модель UML. Существуют также вариации пакетов, например каркасы (Frameworks), модели и подсистемы.

**Аннотационные сущности** – пояснительные части модели UML. Это комментарии для дополнительного описания, разъяснения или замечания к любому элементу модели. Имеется только один базовый тип аннотационных элементов – **примечание** (Note).

**Примечание** – это просто символ для изображения комментариев или ограничений, присоединенных к элементу или группе элементов. **Графически примечание изображается в виде прямоугольника с загнутым краем**, содержащим текстовый или графический комментарий, как показано на рисунке.

http://www.maksakov-sa.ru/files/15.JPG

Этот элемент является основной аннотационной сущностью, которую можно включать в модель UML. Чаще всего примечания используются, чтобы снабдить диаграммы комментариями или ограничениями, которые можно выразить в виде неформального или формального текста. Существуют вариации этого элемента, например требования, где описывают некое желательное поведение с точки зрения внешней по отношению к модели.

В языке UML определены четыре типа отношений:

– зависимость;

– ассоциация;

– обобщение;

– реализация.

Эти отношения являются основными связующими строительными блоками в UML и применяются для создания корректных моделей.

***Зависимость*** (Dependency) – это семантическое отношение между двумя сущностями, при котором изменение одной из них, независимой, может повлиять на семантику другой, зависимой. **Графически зависимость изображается в виде прямой пунктирной линии**, часто со стрелкой, которая может содержать метку.

http://www.maksakov-sa.ru/files/16.JPG

***Ассоциация*** (Association) – структурное отношение, описывающее совокупность связей; связь – это соединение между объектами. Разновидностью ассоциации является агрегирование (Aggregation) – так называют структурное отношение между целым и его частями. Графически ассоциация изображается в виде прямой линии (иногда завершающейся стрелкой или содержащей метку), рядом с которой могут присутствовать дополнительные обозначения, например кратность и имена ролей. На рисунке показан пример отношений этого типа.

http://www.maksakov-sa.ru/files/17.JPG

***Обобщение*** (Generalization) – это отношение «специализация/обобщение», при котором объект специализированного элемента (потомок) может быть подставлен вместо объекта обобщенного элемента (родителя или предка). Таким образом, потомок (Child) наследует структуру и поведение своего родителя (Parent). **Графически отношение обобщения изображается в виде линии с незакрашенной стрелкой**, указывающей на родителя, как показано на рисунке.

http://www.maksakov-sa.ru/files/18.JPG

***Реализация*** (Realization) – это семантическое отношение между классификаторами, при котором один классификатор определяет «контракт», а другой гарантирует его выполнение. Отношения реализации встречаются в двух случаях: во-первых, между интерфейсами и реализующими их классами или компонентами, а во-вторых, между прецедентами и реализующими их кооперациями. Отношение реализации **изображается в виде пунктирной линии с незакрашенной стрелкой**, как нечто среднее между отношениями обобщения и зависимости.

http://www.maksakov-sa.ru/files/19.JPG

Четыре описанных элемента являются основными типами отношений, которые можно включать в модели UML. Существуют также их вариации, например уточнение (Refinement), трассировка (Trace), включение и расширение (для зависимостей).

**Задание 2**

Изучил теоретический материал по теме «Строительные блоки UML».

Поскольку UML описывает системы реального времени, очень важно создать концептуальную модель, а затем действовать постепенно. Концептуальную модель UML можно освоить, изучив следующие три основных элемента:

* UML строительные блоки
* Правила подключения строительных блоков
* Общие механизмы UML

В этой главе описываются все строительные блоки UML. Строительные блоки UML могут быть определены как —

* вещи
* отношения
* диаграммы

**вещи**

**Вещи** являются наиболее важными строительными блоками UML. Вещи могут быть —

* структурная
* поведенческий
* группирование
* annotational

**Структурные Вещи**

**Структурные вещи** определяют статическую часть модели. Они представляют физические и концептуальные элементы. Ниже приведены краткие описания структурных вещей.

**Класс —** класс представляет собой набор объектов, имеющих схожие обязанности.

учебный класс

**Интерфейс —** Интерфейс определяет набор операций, которые определяют ответственность класса.

Интерфейс

**Сотрудничество —** Сотрудничество определяет взаимодействие между элементами.

сотрудничество

**Вариант использования. Вариант** использования представляет собой набор действий, выполняемых системой для конкретной цели.

Случай использования

**Компонент —** Компонент описывает физическую часть системы.

Составная часть

**Узел —** узел может быть определен как физический элемент, который существует во время выполнения.



**Поведенческие вещи**

**Поведенческая вещь** состоит из динамических частей моделей UML. Ниже приведены поведенческие вещи —

**Взаимодействие.** Взаимодействие определяется как поведение, состоящее из группы сообщений, которыми обмениваются элементы для выполнения определенной задачи.

взаимодействие

**Конечный автомат —** Конечный автомат полезен, когда важно состояние объекта в его жизненном цикле. Он определяет последовательность состояний, через которые проходит объект в ответ на события. События являются внешними факторами, ответственными за изменение состояния



**Группировка вещей**

**Группирование** может быть определено как механизм для группировки элементов модели UML вместе. Доступна только одна группировка —

**Пакет —** Пакет — это единственная групповая вещь, доступная для сбора структурных и поведенческих вещей.



**Аннотационные вещи**

**Аннотационные вещи** могут быть определены как механизм для сбора замечаний, описаний и комментариев элементов модели UML. **Примечание.** Это единственная доступная аннотационная вещь. Заметка используется для отображения комментариев, ограничений и т. Д. Элемента UML.

Заметка

**отношения**

**Отношения** — это еще один важнейший строительный блок UML. Он показывает, как элементы связаны друг с другом, и эта связь описывает функциональность приложения.

Есть четыре вида доступных отношений.

**зависимость**

Зависимость — это отношения между двумя вещами, в которых изменение одного элемента также влияет на другой.

зависимость

**ассоциация**

Ассоциация — это набор ссылок, которые связывают элементы модели UML. Он также описывает, сколько объектов принимают участие в этих отношениях.

ассоциация

**Обобщение**

Обобщение может быть определено как отношение, которое связывает специализированный элемент с обобщенным элементом. Это в основном описывает отношения наследования в мире объектов.

Обобщение

**реализация**

Реализация может быть определена как отношение, в котором два элемента связаны между собой. Один элемент описывает некоторую ответственность, которая не реализована, а другой реализует их. Эта связь существует в случае интерфейсов.

реализация

**UML-диаграммы**

Диаграммы UML являются окончательным результатом всего обсуждения. Все элементы, отношения используются для создания полной UML-диаграммы, а диаграмма представляет собой систему.

Визуальный эффект UML-диаграммы является наиболее важной частью всего процесса. Все остальные элементы используются для его завершения.

UML включает в себя следующие девять диаграмм, подробности которых описаны в последующих главах.

**Задание 3**

Ответил на контрольные вопросы.

1. Дайте определение «UML»

Ответ: (Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур

2. Перечислите типы отношений

Ответ:

* Зависимость
* Ассоциация
* Обобщение
* Реализация

3. Перечислите механизмы расширения UML

Ответ: 1. **Ограничения** - позволяют определять новые или изменять существующие правила

2. **Теговые величины** (то же что - Помеченные значения (Tagget Value) - 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 используется в в области [разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), для [моделирования](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [бизнес-процессов](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81), [системного проектирования](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и отображения [организационных структур](https://ru.wikipedia.org/wiki/%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0).

5. Дайте определение «Сущность»

Ответ: **Су́щность** (др. -греч. οὐσία, ὑπόστᾰσις; лат. essentia, substantia) — смысл данной вещи, то, что она есть сама по себе, в отличие от всех других вещей и в отличие от изменчивых (под влиянием тех или иных обстоятельств) состояний вещи.

6. Перечислите строительные блоки UML.

Ответ: Строительные блоки UML могут быть определены как —

* вещи
* Отношения
* Диаграммы

7. Перечислите виды общих механизмов UML.

Ответ:

8. Для чего используются механизмы расширения языка UML.

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