## Стереотипы в UML

В UML стереотип — это элемент модели, который определяет назначение других элементов модели. Вы можете использовать стереотип для уточнения значения элемента модели. Например, вы можете применить стереотип «библиотека» к артефакту, чтобы указать, что это артефакт определенного типа. Стереотипы «вызвать», «создать», «создать экземпляр», «ответственность» и «отправить» можно применить к отношениям использования, чтобы точно указать, как один элемент модели использует другой. Стереотип также можно использовать для описания элемента модели, который отличается по значению или использованию от другого элемента модели.

## Структурные диаграммы в UML

**Диаграмма классов.** Эта диаграмма, наиболее распространенная при разработке ПО, используется для изображения логической и физической структуры системы и показывает ее классы. Она похожа на блок-схему, потому что классы представлены в виде блоков. Эта диаграмма предлагает визуальное представление о различных классах и о том, как они взаимосвязаны. У каждого класса есть три секции:

Верхняя секция: имя класса

Средняя секция: атрибуты класса

Нижняя секция: методы или операции класса

**Диаграмма компонентов.** Также известна как блок-схема компонентов, она показывает логические группы элементов и их взаимосвязи. Другими словами, она дает упрощенное представление о сложной системе, разбивая ее на более мелкие компоненты. Каждый из элементов показан в прямоугольной рамке с названием, написанным внутри. Соединители определяют отношения между различными компонентами.

**Диаграмма объектов.** Часто эта диаграмма используется как способ проверить диаграмму классов на точность. Другими словами, будет ли это работать на практике? Она показывает системные объекты и их взаимосвязи и предлагает лучшее представление о потенциальных недостатках проекта, которые необходимо исправить.

**Диаграмма пакетов.** Этот тип используется, чтобы изобразить зависимости между пакетами, которые составляют модель. Основная цель — показать взаимосвязь между различными крупными компонентами, которые образуют сложную систему.

## Поведенческие диаграммы UML

**Диаграмма последовательности.** Эта визуально привлекательная диаграмма, популярная не только в сообществе разработчиков, хорошо

показывает все типы бизнес-процессов. Она просто раскрывает структуру системы, показывая последовательность сообщений и взаимодействий между операторами и объектами в хронологическом порядке. Диаграммы последовательности отображают простую итерацию и ветвление. Это имеет преимущества для многозадачности.

Диаграмма деятельности. Этот тип изображает пошаговый процесс с четким началом и концом. Это набор операций, которые должны быть выполнены, чтобы достичь цели. Она показывает, как каждое действие ведет к следующему, и как все они связаны. Помимо разработки программного обеспечения, они могут использоваться практически в любой бизнес-среде. Их также называют картированием или моделированием бизнес-процессов. Диаграмма вариантов использования. В этом типе описывается, что делает система, но не то, как она это делает. Вариант использования — это набор событий, которые происходят, когда "оператор" использует систему для завершения процесса. Оператор определяется как кто-либо или что-либо, взаимодействующее с системой (человек, организация или приложение) из-за пределов системы. Таким образом, диаграмма вариантов использования визуально описывает этот набор последовательностей и представляет функциональные требования системы.

## Виды отношений между классами в UML

Отношение ассоциации. Отношение ассоциации используют, чтобы показать, что между классами (например, между двумя классами) существует некоторая связь. Обычно с помощью него на диаграмме классов показывают, что один класс пользуется функционалом другого класса. Стрелка ассоциации направлена от класса пользователя к классу владельцу используемой функциональности. Для пояснения того, каким образом один класс использует другой класс, вы можете описать данный процесс в вспомогательном тексте.

**Отношение зависимости.** Отношение зависимости используют, чтобы показать, что изменение одного класса требует изменение другого класса. Стрелка отношения зависимости направлена от зависимого класса к независимому.

Отношение наследования. Отношение наследования используется, чтобы показать, что один класс является родителем (базовым классом или суперклассом) для другого класса (потомка, производного класса). Отношение агрегации. Отношение агрегации между двумя классами показывает, что один из них включает в себя другой класс в качестве составной части.

**Отношение композиции.** Отношение композиции является частным случаем отношения агрегации. Однако у него есть одно отличие — классычасти, которые он соединяет с классом-целым, не могут существовать обособленно.