Диаграмма состояний (State diagram) — это один из видов диаграмм UML, используемых в разработке программного обеспечения, чтобы визуализировать и моделировать поведение объекта или системы в различных состояниях. Она позволяет описать все возможные состояния объекта, а также переходы между ними в ответ на определенные события.

Диаграмма состояний описывает все возможные состояния, в которых может находиться объект или система, а также переходы между этими состояниями. Она позволяет визуально представить различные состояния объекта или системы и показать, как они изменяются в ответ на события или внешние условия. Переходы обозначают изменение состояния и указывают, при каких условиях происходит переход между состояниями. События являются внешними сигналами или действиями, которые вызывают переходы между состояниями.

На диаграмме состояний UML состояния обычно представляются в виде прямоугольников с названиями, а переходы — стрелками. Дополнительные аннотации могут использоваться для указания условий переходов или действий, выполняемых в определенных состояниях.

Диаграмма состояний UML широко используется при анализе и проектировании систем, где важно моделировать поведение объектов или системы в различных сценариях. Они помогают разработчикам лучше понять и определить логику состояний и переходов, что способствует более эффективной реализации и тестированию программного обеспечения.

**Компоненты диаграммы состояний**

**Начальное состояние** (Start state)



Начальное состояние — это состояние, с которого начинается выполнение процесса.

Оно обозначается с помощью символа заполненного кружка, откуда исходят стрелки, представляющие переходы в другие состояния.

Начальное состояние указывает на точку входа в модель поведения и определяет, с какого состояния объект или система начинает свое выполнение или реагирует на события. Когда диаграмма состояний запускается, она сразу переходит из начального состояния в другое состояние в ответ на определенное событие или действие.

Пример отображения начального состояния на диаграмме состояний

Пример отображения начального состояния на state diagram

Обычно начальное состояние не содержит никакой логики или действий, оно служит для обозначения стартовой точки.

Однако оно может быть связано с переходами, которые определяют первое состояние,

к которому будет осуществлен переход после входа в систему или после выполнения некоторых начальных действий.

Начальное состояние может быть единственным на диаграмме состояний или использоваться несколько начальных состояний,

чтобы указать на возможность параллельного или неопределенного старта в различных состояниях системы.

**Конечное состояние** (Final state)



Конечное состояние — это состояние, в котором процесс завершает свое выполнение.

Оно обозначается обычно с помощью символа закрытого кружка.

Конечное состояние указывает на завершение объекта или системы и обозначает,

что дальнейшее выполнение или процесс достиг своего завершения. При достижении конечного состояния объект или система останавливается, и дальнейшие переходы или события становятся недопустимыми.

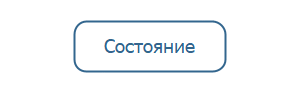
Конечное состояние может быть одним или множественным на диаграмме состояний. Если на диаграмме присутствует только одно конечное состояние, то оно указывает на окончательное завершение объекта или системы. Если же присутствуют несколько конечных состояний, это может означать возможность завершения в различных конечных состояниях, которые могут быть достигнуты в зависимости от определенных условий или событий.

Пример отображения конечного состояния на диаграмме состояний

Обозначение конечного состояния на state diagram

Конечное состояние не имеет исходящих переходов, после его достижения объект не переходит в другие состояния. Однако переходы, ведущие к конечному состоянию, могут быть определены из других состояний на диаграмме.

**Состояние** (State)



Состояние в диаграмме состояний UML представляет определенное состояние объекта или системы, которое может изменяться в ответ на определенные события, условия или действия. Оно определяет его поведение и свойства в определенный момент времени.

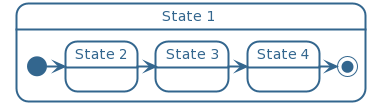
В диаграмме состояний UML состояние обычно представляется прямоугольником с названием состояния. Состояния могут быть обозначены текстовыми метками, описывающими их смысл или функциональность. Например, состояние «ожидание», «активный», «заблокирован» и т.д.

Пример отображения состояния на диаграмме состояний

Обозначение состояния на state diagram

Состояние объекта может быть долговременным или мгновенным. Долговременное состояние означает, что объект остается в этом состоянии на протяжении некоторого времени, пока не произойдет событие или условие, вызывающее переход в другое состояние. Мгновенное состояние, с другой стороны, представляет собой непродолжительный момент времени, когда объект находится в определенном состоянии перед выполнением перехода.

**Составное состояние** (Composite state)



Составное состояние в диаграмме состояний UML представляет собой состояние, которое содержит внутренние состояния и переходы между ними. Оно позволяет структурировать более сложные состояния и их поведение внутри объекта или системы.

Составное состояние обычно представляется в виде прямоугольника, внутри которого содержатся другие состояния и переходы. Внутренние состояния и переходы могут быть представлены как отдельные элементы внутри составного состояния или в виде вложенных диаграмм состояний.

В составном состоянии можно определить иерархию вложенных состояний, которая отображает различные уровни детализации поведения объекта или системы. Когда объект находится в составном состоянии, его поведение может зависеть от текущего внутреннего состояния и переходить между ними в ответ на определенные события или условия.

Пример отображения составного состояния на state diagram

Обозначение составного состояния на диаграмме состояний

Составные состояния обычно используются для моделирования сложных сценариев или поведения системы, где объект может находиться в различных подсостояниях и переходить между ними в соответствии с определенными правилами или условиями. Они позволяют более структурировано и понятно представить сложные потоки выполнения и взаимодействия объектов в системе.

**Защитное условие** (Guard Condition)

Защитное условие представляет собой логическое выражение, которое определяет условия, при которых может произойти переход из одного состояния в другое. Оно является частью перехода и помогает управлять потоком выполнения между состояниями.

Защитное условие указывает на необходимость удовлетворения определенного условия для совершения перехода. Если условие истинно (выполняется), переход может произойти. Если условие ложно, переход будет заблокирован, и объект или система останется в текущем состоянии.

Защитные условия могут быть выражены с помощью логических операторов и сравнений для проверки значений переменных, флагов или других условий, которые влияют на переходы между состояниями.

Защитные условия позволяют управлять переходами между состояниями на основе определенных условий, что делает модель более гибкой и реалистичной. Они позволяют объекту принимать решения о переходах, основываясь на текущих условиях и переменных окружения.

**Разделитель** (Concurrent state)

Разделитель, также известный как параллельное состояние или состояние с одновременным выполнением, представляет состояние, в котором объект или система может находиться одновременно в нескольких взаимодействующих состояниях. Это позволяет моделировать параллельное выполнение или конкурентные аспекты поведения объекта или системы.

Разделитель представляется горизонтальной линией, пересекающей состояния и переходы. Он указывает, что объект может находиться в различных состояниях одновременно и выполнять соответствующие действия параллельно.

Состояния, расположенные выше и ниже разделителя, являются параллельными и могут выполняться независимо друг от друга. Переходы, связанные с каждым параллельным состоянием, могут срабатывать независимо в зависимости от условий или событий.

Разделитель часто используется для моделирования параллельных процессов или параллельных потоков выполнения в системе. Он позволяет представить ситуации, когда объект выполняет несколько действий одновременно или когда в системе происходит взаимодействие между различными компонентами, выполняющими разные задачи одновременно.

**Историческое состояние** (Historical State)



Историческое состояние представляет специальное состояние, которое сохраняет информацию о предыдущем состоянии объекта или системы. Оно используется для моделирования возврата к предыдущему состоянию или сохранения контекста состояния.

В диаграмме состояний историческое состояние обычно обозначается символом «H» внутри состояния. История может иметь входящие и исходящие переходы, позволяющие объекту или системе восстановить предыдущее состояние или перейти к новому состоянию в зависимости от контекста.

Историческое состояние историческое состояние сохраняет только само состояние, без сохранения информации о подсостояниях. При переходе через такое состояние объект возвращается к предыдущему состоянию без восстановления его внутренних состояний.

Пример отображения исторического состояния на диаграмме состояний

Обозначение исторического состояния на диаграмме состояний

Использование истории состояний позволяет объекту возвращаться к предыдущему состоянию и продолжать выполнение соответствующего поведения. Это полезно в ситуациях, когда после выполнения определенного действия или события объект должен вернуться к состоянию, которое он находил ранее и продолжая выполнение с того места, где был оставлен.

**Глубокое историческое состояние** (Deep Historical State)



Глубокое историческое состояние является особой формой истории состояний, которая сохраняет информацию о предыдущем состоянии объекта и всех его внутренних состояниях. Оно обозначается символом «H\*» внутри состояния. Оно имеет входящие и исходящие переходы, позволяющие объекту восстановить предыдущее состояние со всеми вложенными состояниями.

Пример отображения глубокого исторического состояния на state diagram

Обозначение глубокого исторического состояния на диаграмме состояний

Использование глубокой историй полезно в ситуациях, когда объект имеет сложную структуру состояний с вложенными подсостояниями. При переходе через глубокую историю объект возвращается в предыдущее состояние вместе с его подсостояниями.

**Переход** (Transition)



Переход — это элемент диаграммы состояний, который представляет собой переход объекта из одного состояния в другое. Он определяет событие или условие, которое вызывает изменение состояния, и указывает, какой переход должен быть выполнен при наступлении этого события или условия.

Переходы обычно представляются стрелками, которые соединяют состояния в диаграмме состояний. Они могут иметь метку, которая указывает событие или условие, при котором переход может быть выполнен.

Переходы могут быть направленными или ненаправленными. Направленный переход указывает однонаправленный поток выполнения от одного состояния к другому. Ненаправленный переход представляет переход, который может быть выполнен в обоих направлениях между состояниями.

Обозначение перехода на state diagram

Обозначение перехода на диаграмме состояний

При наступлении события или удовлетворении условия, связанного с переходом, объект или система переходит из текущего состояния в целевое состояние, которое указано на конце перехода. В результате выполнения перехода могут быть выполнены действия или изменения внутреннего состояния объекта.

Переходы позволяют моделировать потоки выполнения и изменения состояний объекта, отражая его поведение и реакцию на события или условия. Они помогают создать более детализированную и понятную модель объекта или системы в рамках диаграммы состояний.