

Диаграмма прецедентов представляет собой простой способ визуального представления основных возможностей разрабатываемого программного обеспечения или процесса.

Пример диаграммы прецедентов, описывающий функции текстового редактора:

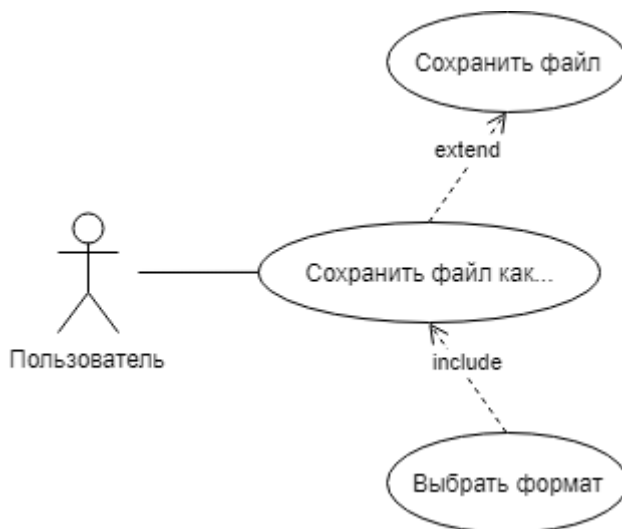


Для описания функций используются действующее лицо (обозначенное человекоподобной фигурой), прецеденты (обозначенные овалами) и ассоциативные связи.

Следует отметить, что действующим лицом может являться не только пользователь, но и программа, программист, специалист службы технической поддержки и т.д. А на одной диаграмме может присутствовать множество действующих лиц.

Помимо ассоциативных связей существуют:

- направленные ассоциации (линия со стрелкой) – в явной форме указывают характер отношений между прецедентами;
- зависимости – указывают на зависимости между прецедентами;
- обобщения – указывают на вхождение частного прецедента в более общий.

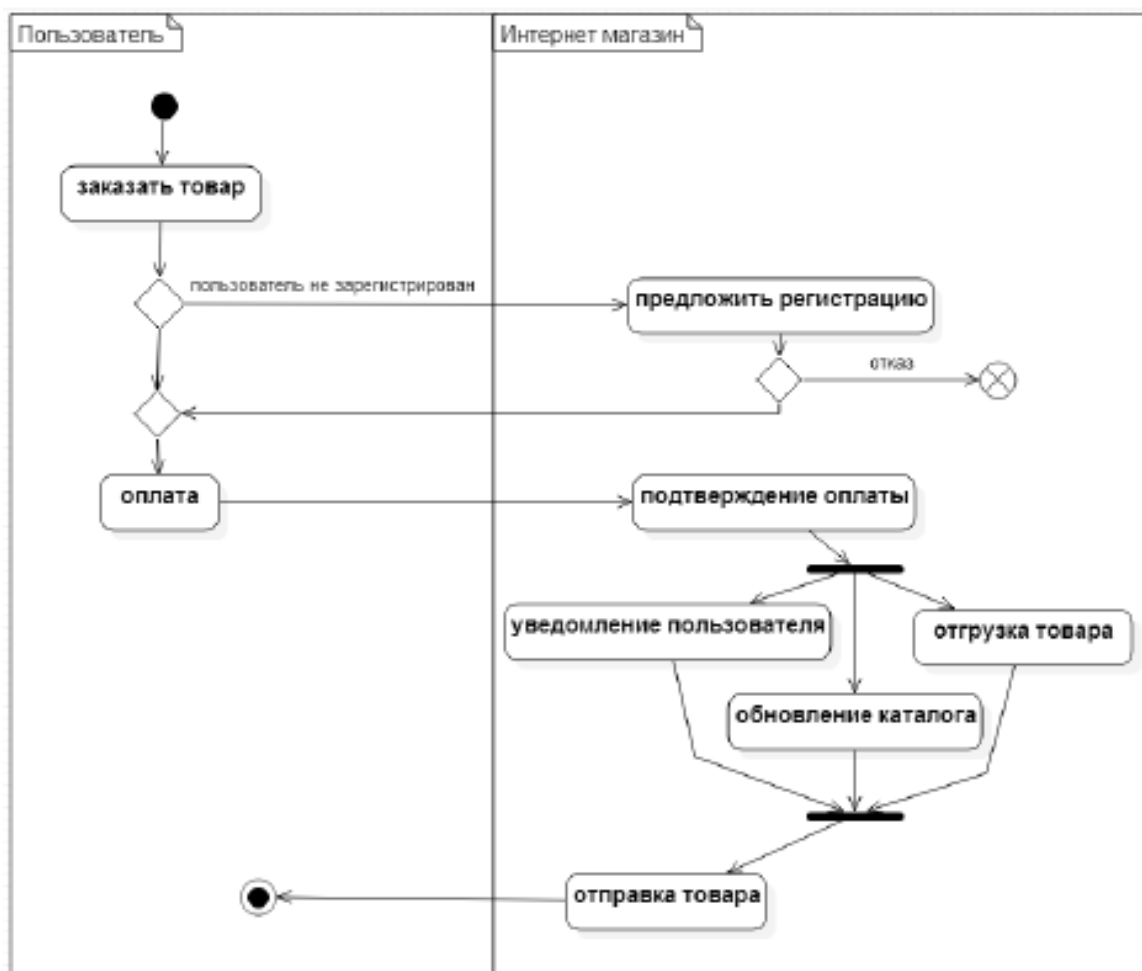


В данном примере описан прецедент расширенной функции «сохранить файл». Для указания того, что сохранение файла с выбором формата содержит в себе выбор формата и является расширением стандартной функции сохранения файла, используется отношение зависимости.



На данной диаграмме показано, что ввод текста является частным случаем редактирования файла, а ввод связан с выбором шрифта (без указания типа связи). Следует помнить, что диаграмма прецедентов должна описывать общие функциональные возможности разрабатываемой программы или системы, а так же фундаментальные зависимости между прецедентами, не акцентируя внимания на деталях реализации процессов и действующих лиц.

Диаграмма деятельности является альтернативой представлению процессов в виде блок-схем и используется для описания последовательности действий и выборов. Состоит из следующих элементов:



- начало процесса – обозначает старт описываемого процесса, может не совпадать с началом работы программы или глобального процесса;
- действие – содержит в себе описание действий на текущем этапе выполнения алгоритма;
- решение – как и на блок-схемах обозначается ромбом, однако не содержит в себе текста. Текст условий ветвления указывается на исходящих из решения управляющих потоках;
- управляющий поток – указывает последовательность выполнения действий;
- разделение – начало блока независимых операций;

- соединение – завершение блока независимых операций;
- завершение процесса – окончание описываемого процесса, может не совпадать с окончанием работы программы или глобального процесса.

Выше изображен пример описания процесса заказа товара через интернет-магазин при помощи диаграммы деятельности. На представленной диаграмме действия расположены в двух областях, обозначающих действующих лиц, участвующих в процессе заказа. Подобное представление не является обязательным при составлении диаграммы деятельности, однако, в дальнейшем, может упростить создание диаграммы последовательности. Предполагается что операции по уведомлению пользователя, обновлению каталога и отгрузки товара, могут осуществляться одновременно, поскольку независимы друг от друга. Действие «отправка товара» является обобщённым и может быть представлено в виде отдельной диаграммы, если это необходимо для понимания моделируемых процессов. Следует помнить, что диаграмма деятельности должна описывать последовательность действий и выборов в процессе выполнения некоего процесса, не акцентируя внимания на классах, полях и методах.

Диаграмма классов - диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов, методов, интерфейсов и взаимосвязей между ними. Широко применяется для документирования и визуализации информационных систем.

Основной структурной сущностью в диаграммах классов является класс. **Класс** – это описание набора объектов с одинаковыми атрибутами, операциями, связями и семантикой. Графически класс изображается в виде прямоугольника, разделенного на 3 блока горизонтальными линиями:

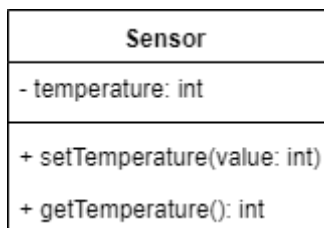
- имя класса;
- атрибуты (свойства) класса;
- операции (методы) класса.

Каждый класс должен обладать именем, отличающим его от других классов. Обычно используются краткие имена, отражающие описываемую сущность. Каждое слово в имени класса традиционно пишут с заглавной буквы, например Sensor или Датчик.

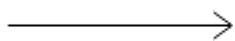
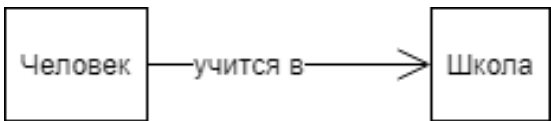
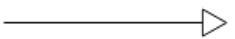

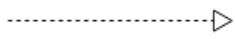
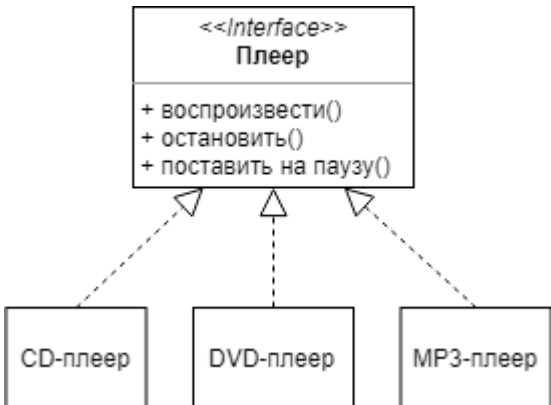
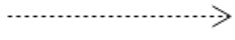
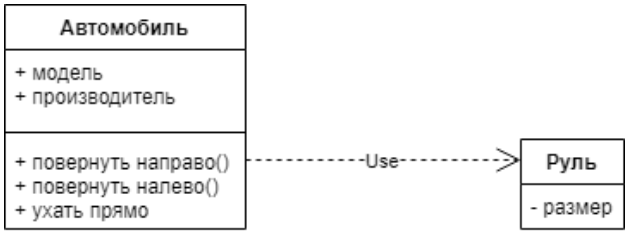

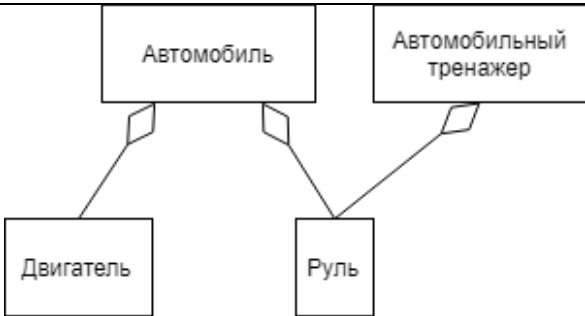
Атрибут (свойство) представляет некоторое свойство моделируемой сущности, которым обладают все объекты данного класса. Класс может иметь любое число атрибутов или не иметь ни одного. В последнем случае блок атрибутов оставляют пустым.


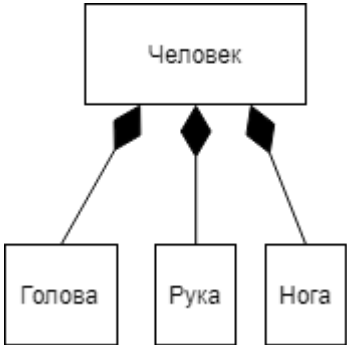
Операция (метод) – это реализация метода класса. Класс может иметь любое число операций либо не иметь ни одной. Часто вызов операции объекта изменяет его атрибуты. Класс может иметь любое число операций либо не иметь ни одной. Графически операции представлены в нижнем блоке описания класса. Можно специфицировать операцию, устанавливая ее сигнатуру, включающую имя, тип и значение по умолчанию всех параметров, а применительно к функциям – тип возвращаемого значения.

Изображая класс, не обязательно показывать сразу все его атрибуты и операции. Для конкретного представления, как правило, существенна только часть атрибутов и операций класса. В силу этих причин допускается упрощенное представление класса, то есть для графического представления выбираются только некоторые из его атрибутов. Если помимо указанных существуют другие атрибуты и операции, вы даете это понять, завершая каждый список многоточием.



Взаимосвязь (отношение) - это особый тип логических отношений между сущностями, показанных на диаграммах классов и объектов. В UML представлены следующие виды отношений:

Имя взаимосвязи и графическое обозначение	Описание	Пример
Ассоциация 	показывает, что объекты одной сущности (класса) связаны с объектами другой сущности таким образом, что можно перемещаться от объектов одного класса к другому	
Наследование 	показывает, что один из двух связанных классов (подтип) является частной формой другого (надтипа)	
Реализация/Имплементация 	отношение между классами, в котором один из них (клиент) реализует поведение, заданное другим (поставщиком)	
Зависимость 	связь использования, указывающая, что изменение спецификаций одной сущности может повлиять на другие сущности, которые используют её	
Агрегация 	особая разновидность ассоциации, представляющая структурную связь целого с его частями	

<p>Композиция</p> 	<p>форма агрегации с четко выраженными отношениями владения и совпадением времени жизни частей и целого</p>	 <pre> classDiagram class Человек class Голова class Рука class Нога Человек o-- Голова Человек o-- Рука Человек o-- Нога </pre>
---	---	--

Список литературы:

- 1) Введение в UML: <https://www.intuit.ru/studies/courses/1007/229/info>