

Визуальное моделирование поведения программной системы на основе вариантов использования UML



UML (Unified Modeling Language)

Язык UML представляет собой общецелевой язык визуального моделирования, который разработан для спецификации, визуализации, проектирования и документирования компонентов программного обеспечения, бизнес-процессов и других систем.

Унифицированный язык моделирования UML является графическим языком для визуализации, конструирования и документирования систем, в которых большая роль принадлежит программному обеспечению.

Сайт <http://www.uml.org/>

Посмотреть историю изменения версий UML : <http://www.uml-diagrams.org/>

Скачать спецификации со страницы: <http://www.omg.org/spec/UML/2.4/>

Модель сложной системы в нотации UML



Важность использования диаграммы прецедентов

Идентификация функций и как с ними взаимодействуют роли – основное назначение диаграмм сценариев использования.

Для представления системы на высоком уровне – при представлении ее руководителям или заинтересованным сторонам.

Выделяют роли, которые взаимодействуют с системой, и функциональные возможности, предоставляемые системой, не углубляясь во внутреннюю работу системы.

Диаграммы прецедентов = вариантов использования

Описывает функциональное назначение системы, т.е. то, что система будет делать в процессе своего функционирования;
Является *исходной концептуальной моделью системы* в процессе ее проектирования и разработки

Цели построения диаграмм вариантов использования

- 1) определить **общие границы** и **контекст** моделируемой предметной области на начальных этапах проектирования;
- 2) сформулировать **общие требования** к функциональному проектированию системы;
- 3) разработать **исходную концептуальную модель** системы для ее последующей реализации;
- 4) подготовить **документацию** для взаимодействия *разработчика* системы с ее *заказчиком* и *пользователями*.

Суть диаграммы прецедентов

Проектируемая система представляется в виде множества сущностей или экторов (действующих лиц), взаимодействующих с системой с помощью так называемых вариантов использования (прецедентов).

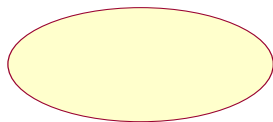
Основными компонентами ДВИ являются:

1. актеры
2. прецеденты
3. отношения

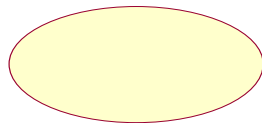
Вариант использования

= Прецедент = use case = юскейс;

Определяет последовательность действий, которая должна быть выполнена проектируемой системой при взаимодействии ее с соответствующим актером.



Получение справки о
состоянии счета



Получить справку о
состоянии счета



(а)

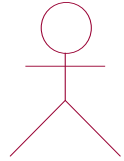


(б)

Имя ВИ начинается с большой буквы и обозначается
оборотом глагола или существительного,
обозначающего действие

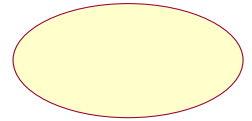
Use-case diagram

1) Actor (участник) — множество логически связанных ролей, исполняемых при взаимодействии с прецедентами или сущностями (система, подсистема или класс). Участником может быть человек, роль человека в системе или другая система, подсистема или класс, которые представляют нечто вне сущности.



Клиент банка

2) Use case (прецедент) — описание отдельного аспекта поведения системы с точки зрения пользователя. Прецедент не показывает, "как" достигается некоторый результат, а только "что" именно выполняется.



Получение справки о состоянии счета

Правила построения

1. каждый прецедент относится как минимум к одному действующему лицу;
2. каждый прецедент имеет инициатора;
3. каждый прецедент приводит к соответствующему результату.

Use-case diagram

Два участника: студент и библиотекарь.

Прецеденты для студента: ищет в каталоге, заказывает, работает в читальном зале.

Роль библиотекаря: выдача заказа, консультации (рекомендации книг по теме, обучение использованию поисковой системы и заполнению бланков заказа).



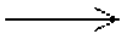
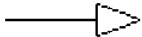
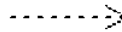

Отношения

Один актер может взаимодействовать с несколькими вариантами использования и наоборот.

Два варианта использования, определенные для одной и той же сущности, не могут взаимодействовать друг с другом, т.к. любой из них самостоятельно описывает законченный вариант использования этой сущности.

Виды отношений

- 1) ассоциативное отношение (отношение ассоциации, association relationship)
- 2) отношение обобщения (generalization relationship)
- 3) отношение включения (include relationship)
- 4) отношение расширения (extend relationship)

Обозначение	Отношение
	Связь коммуникации (communication)
	Связь обобщения (generalization)
	Связь включения (include)
	Связь расширения (extend)

Отношение ассоциации

Отношение между вариантом использования и актером, отражающее связь между ними.

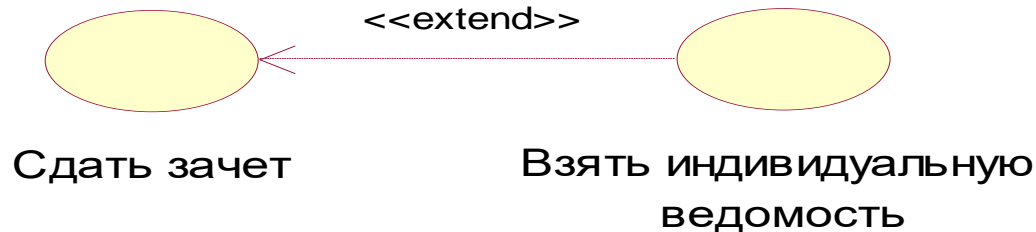
Оно устанавливает, какую конкретную роль играет актер при взаимодействии с экземпляром варианта использования.



Обозначение: в виде прямой линии. Могут быть дополнительные обозначения (кратность связи, направление связи, наименование связи)

Отношение расширения

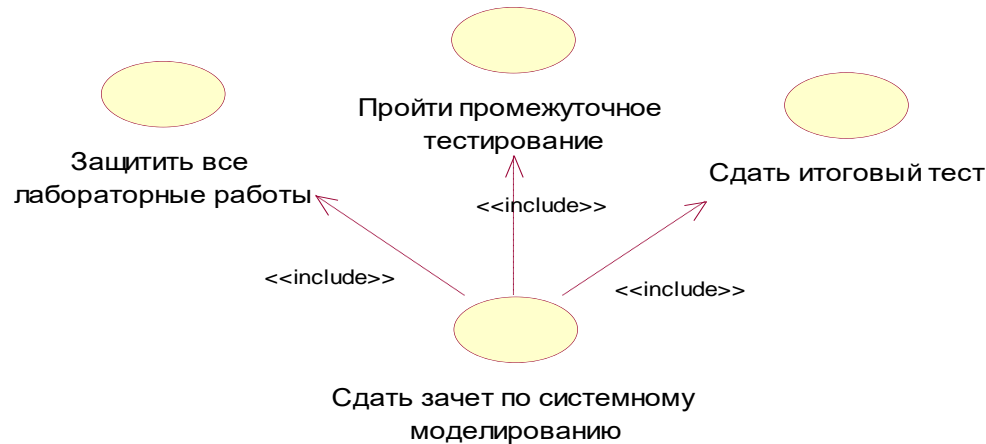
Определяет взаимосвязь базового варианта использования с некоторым другим вариантом использования, функциональное поведение которого задействуется базовым не всегда, а только при выполнении некоторых дополнительных условий.



Стрелка указывает на базовый вариант использования!

Отношение включения

Указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования.



Как найти варианты использования

1. Прочитать документацию заказчика.
2. Рассмотреть области использования системы.
3. Учесть мнение каждого из заинтересованных лиц проекта.
4. Учесть реакцию системы на внешние события.

Как найти варианты использования

1. Присутствует ли каждое функциональное требование хотя бы в одном варианте использования?
2. Как с системой будет работать каждое заинтересованное лицо?
3. Какую информацию каждое заинтересованное лицо будет передавать системе?
4. Какую информацию каждое заинтересованное лицо будет получать от системы?
5. Кто-то должен будет запускать готовую систему и выключать ее при эксплуатации
6. Учли ли вы все внешние системы, с которыми будет взаимодействовать данная? и т.д.

Диаграмма прецедентов для процесса постройки дома

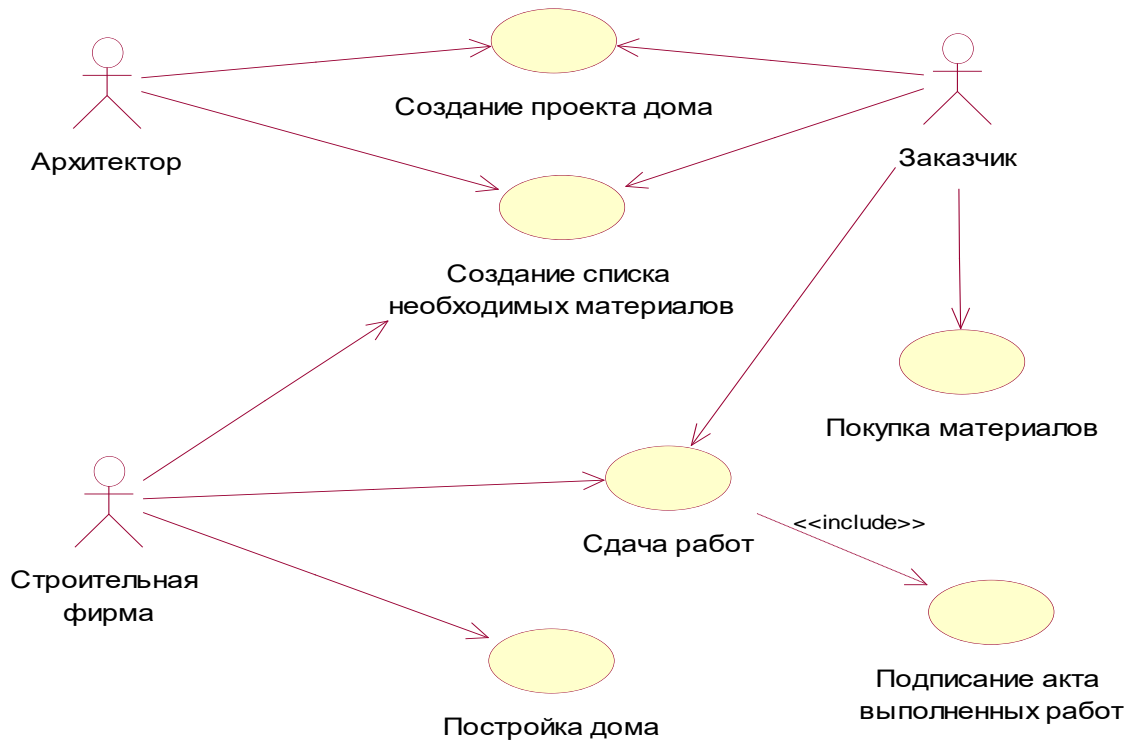
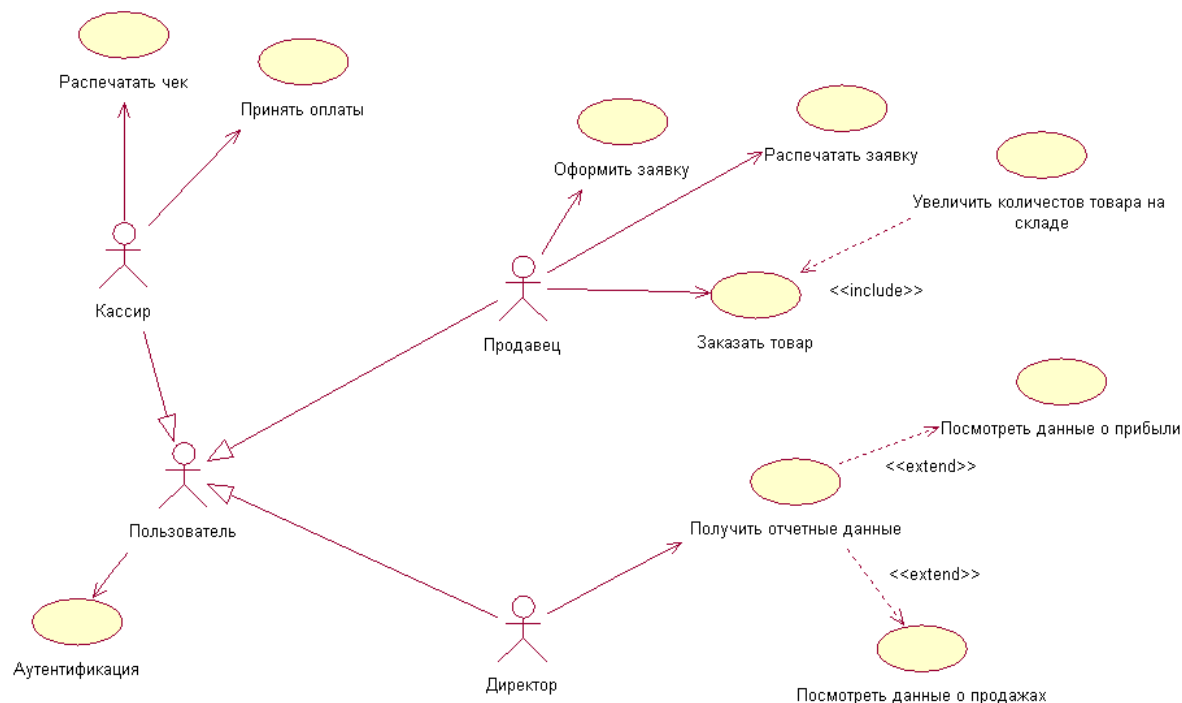


Диаграмма прецедентов для магазина



Диаграммы деятельности

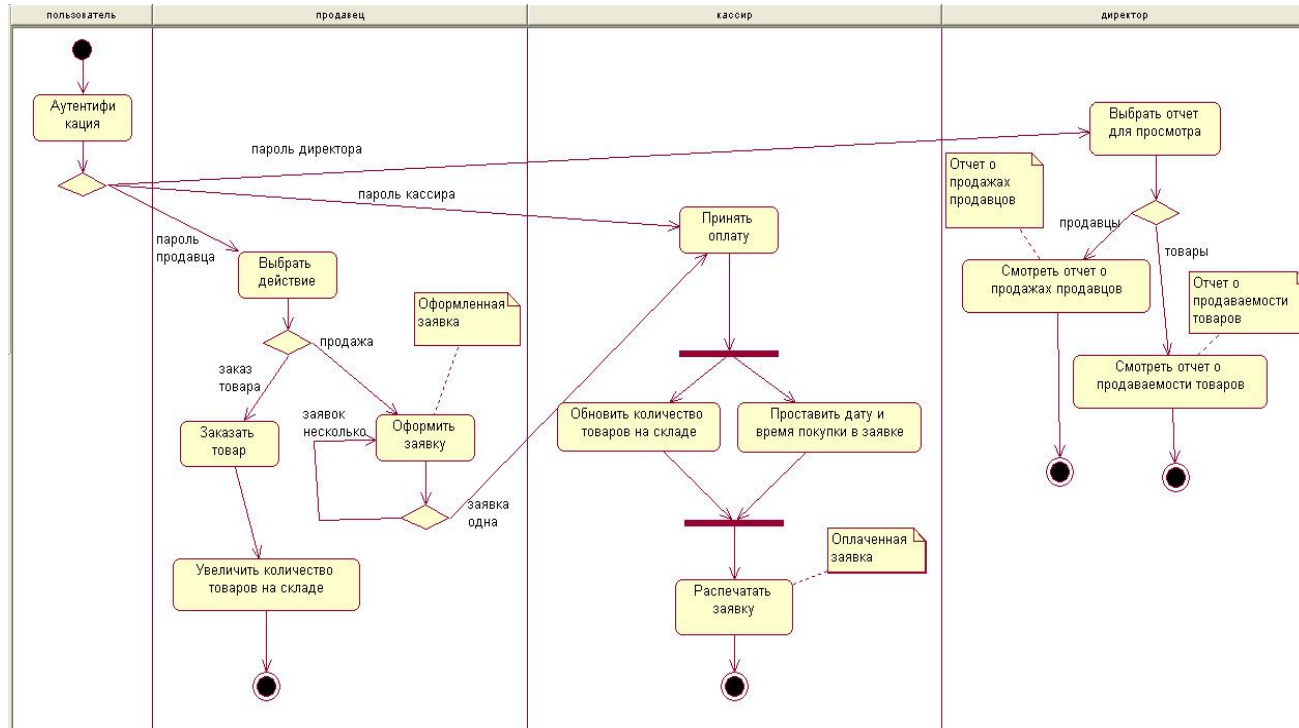
Диаграмма деятельности показывает, как поток управления переходит от одной деятельности к другой.

Диаграмма деятельности в общем случае содержит

- действия,
- переходы,
- объекты.

В языке UML деятельность (activity) представляет собой совокупность отдельных вычислений, приводящих к некоторому результату или действию (action).

Диаграммы деятельности



Диаграммы деятельности

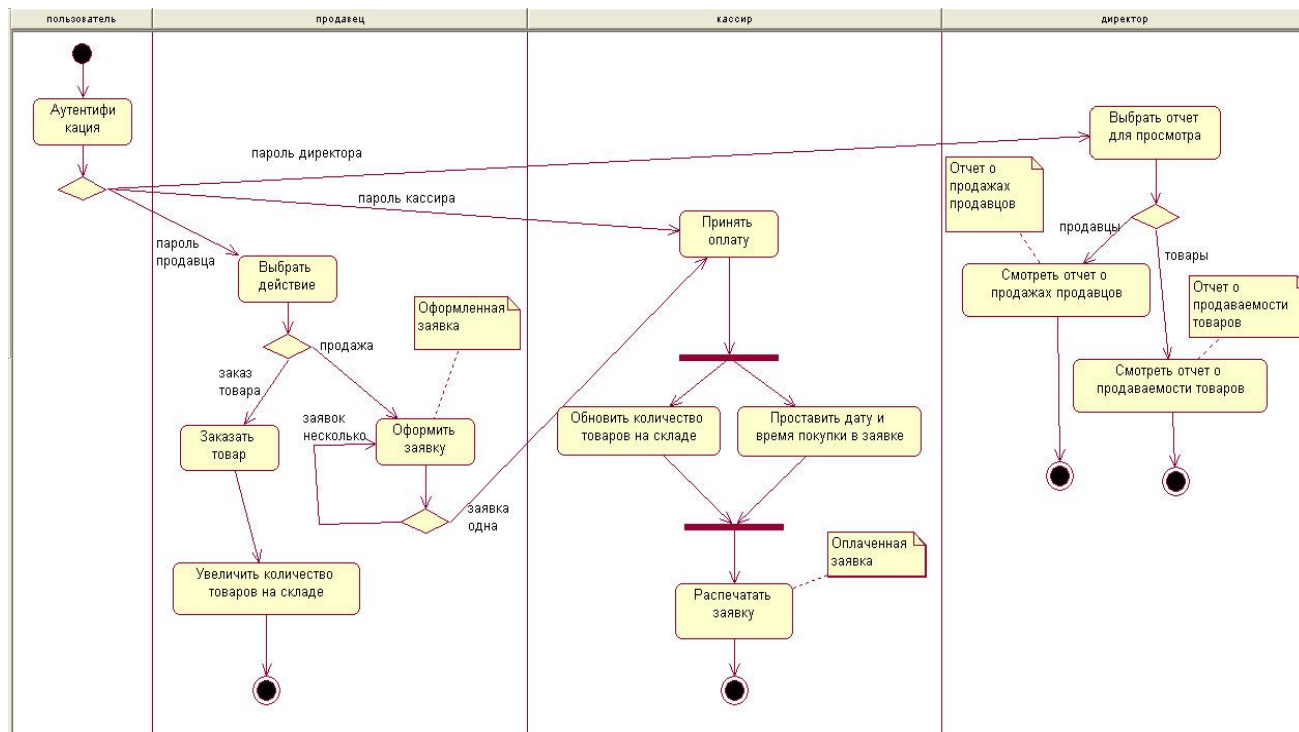


Диаграмма последовательности

Диаграммой последовательности называется диаграмма взаимодействий, акцентирующая внимание на упорядоченности сообщений во времени.

Графически такая диаграмма представляет собой таблицу, объекты в которой располагаются вдоль оси X, а сообщения в порядке возрастания времени - вдоль оси Y.

Как правило, диаграммы последовательностей содержат

- объекты
- связи
- сообщения.

Диаграммы последовательности

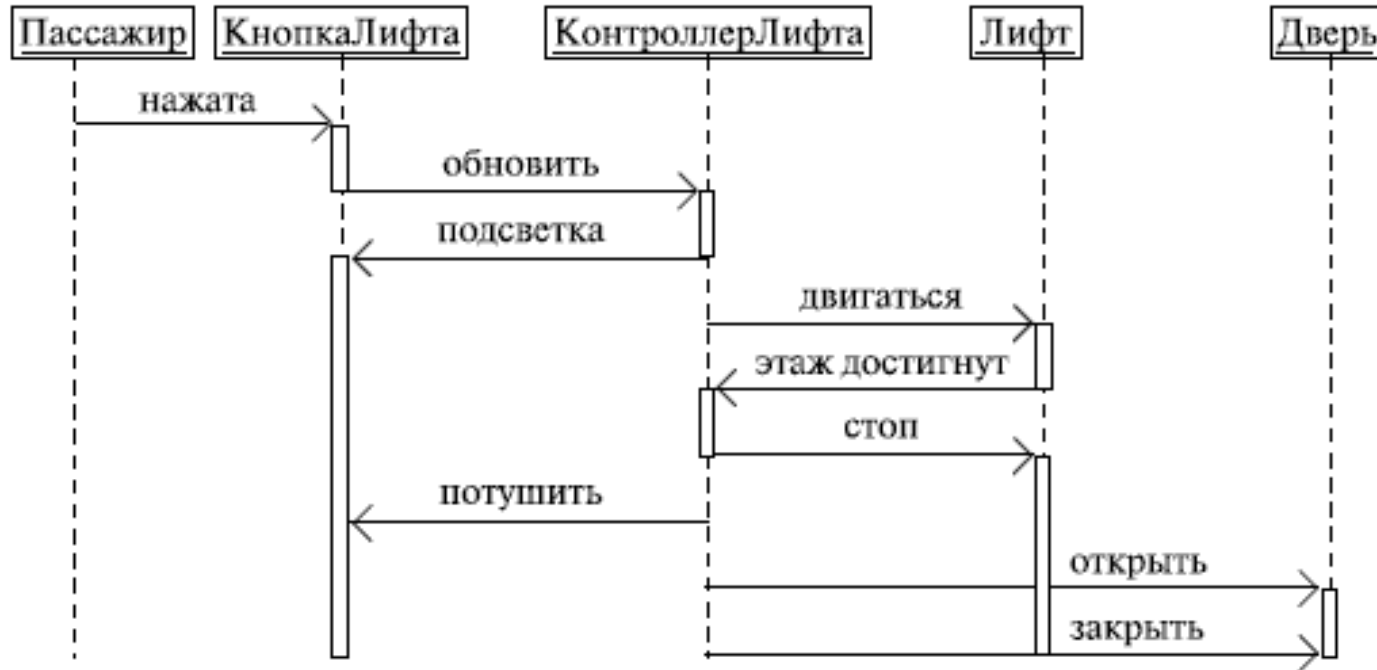


Диаграмма последовательности

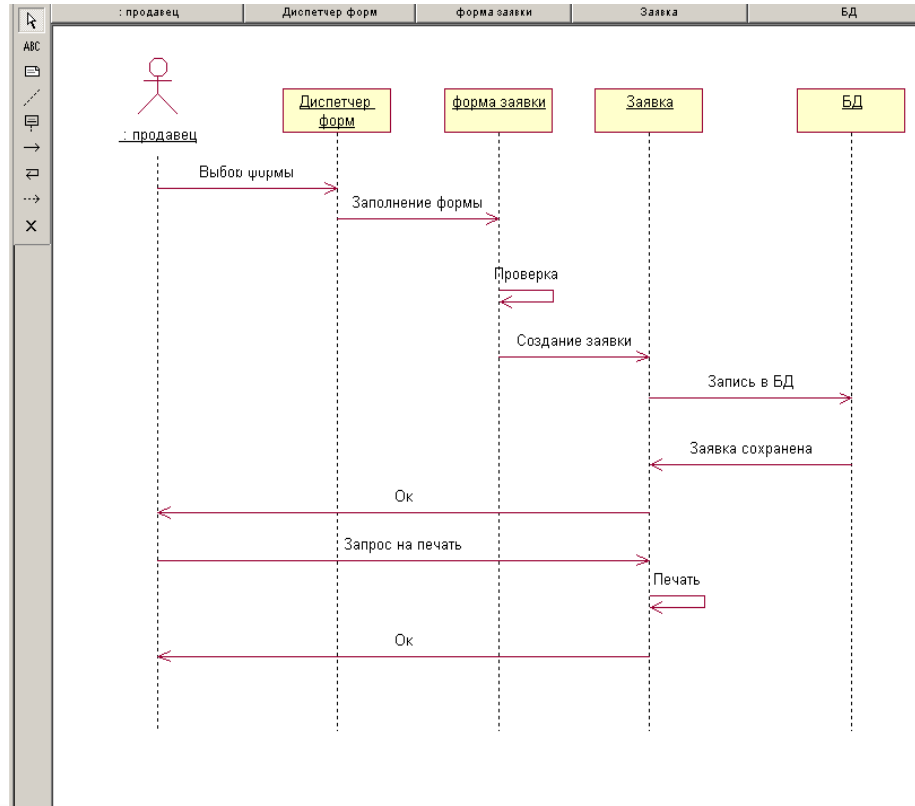
Взаимодействие объектов - обмен информацией между ними.
Информация принимает вид сообщений.

Диаграмма последовательностей отображает взаимодействие объектов в динамике, во времени.

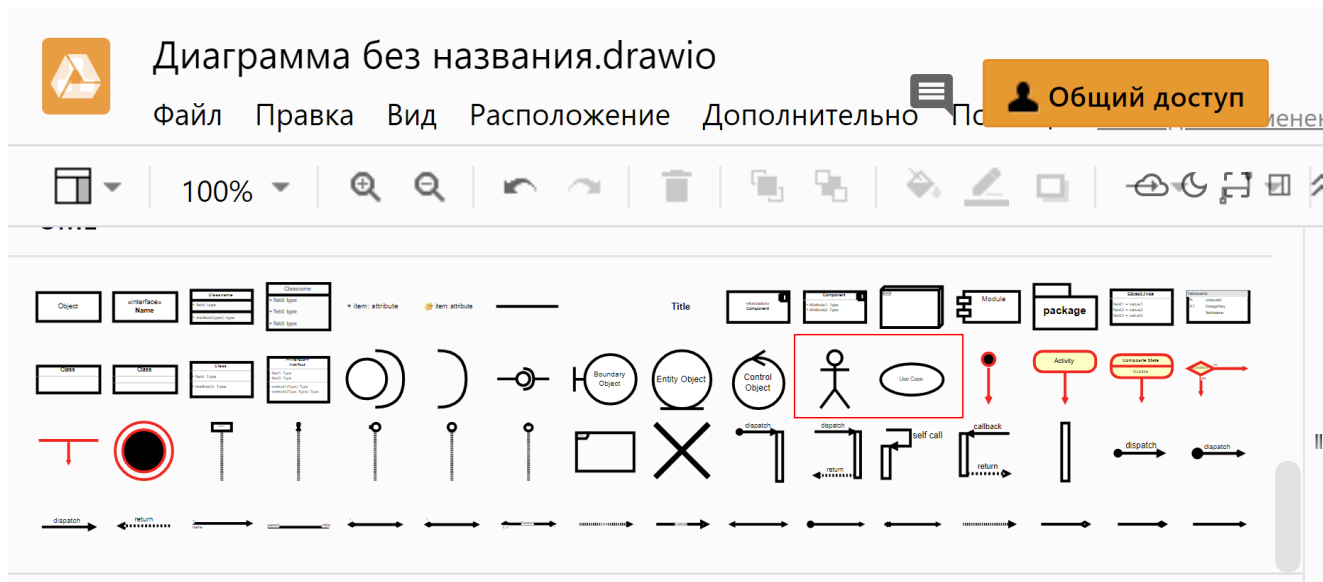
Отражает **поведенческие** аспекты системы

Отображает временные особенности передачи и приема сообщений объектами.

Диаграммы последовательности



Программные среды для реализации UML-диаграмм



<https://app.diagrams.net/>

<https://creately.com/diagram-community/examples/t/use-case>

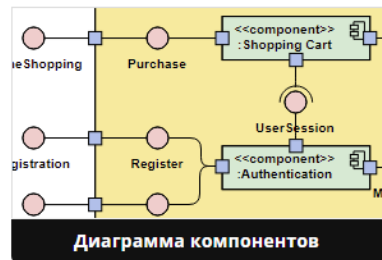
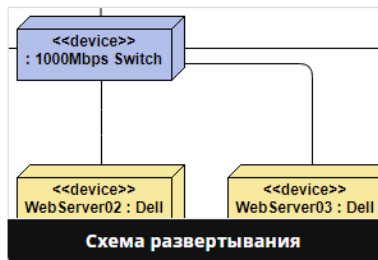
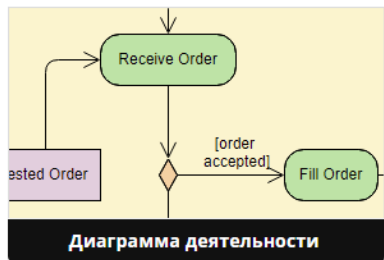
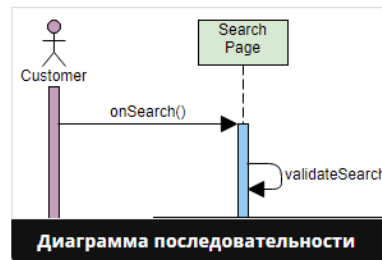
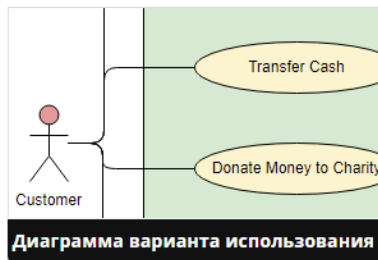
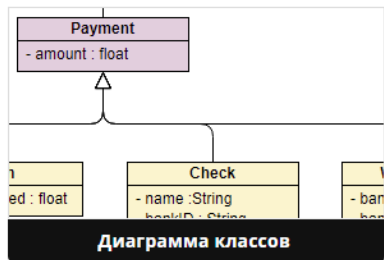
The first diagram shows a general UML Use Case Diagram template. It features a central box labeled 'System' containing seven orange oval use cases, each labeled 'Use Case'. To the left, a blue stick figure actor is connected to all seven use cases. To the right, three stick figure actors (green, red, and blue) are connected to specific use cases: the green actor to the top two, the red actor to the middle two, and the blue actor to the bottom three. A yellow stick figure actor is also shown on the right but is not connected to any use cases.

The second diagram, titled '<I.I Use Case Name> for System X', shows a 'Simple UseCase Template'. It features a central box labeled 'System X' containing seven blue oval use cases. The use cases are arranged in a hierarchical structure: '1. Use Case' and '2. Use Case' are connected to a green stick figure actor on the left; '3. Use Case' and '4. Use Case' are connected to a green stick figure actor on the right; '5. Use Case' and '6. Use Case' are connected to a green stick figure actor on the left; and '7. Use Case' and '8. Use Case' are connected to a green stick figure actor on the right.

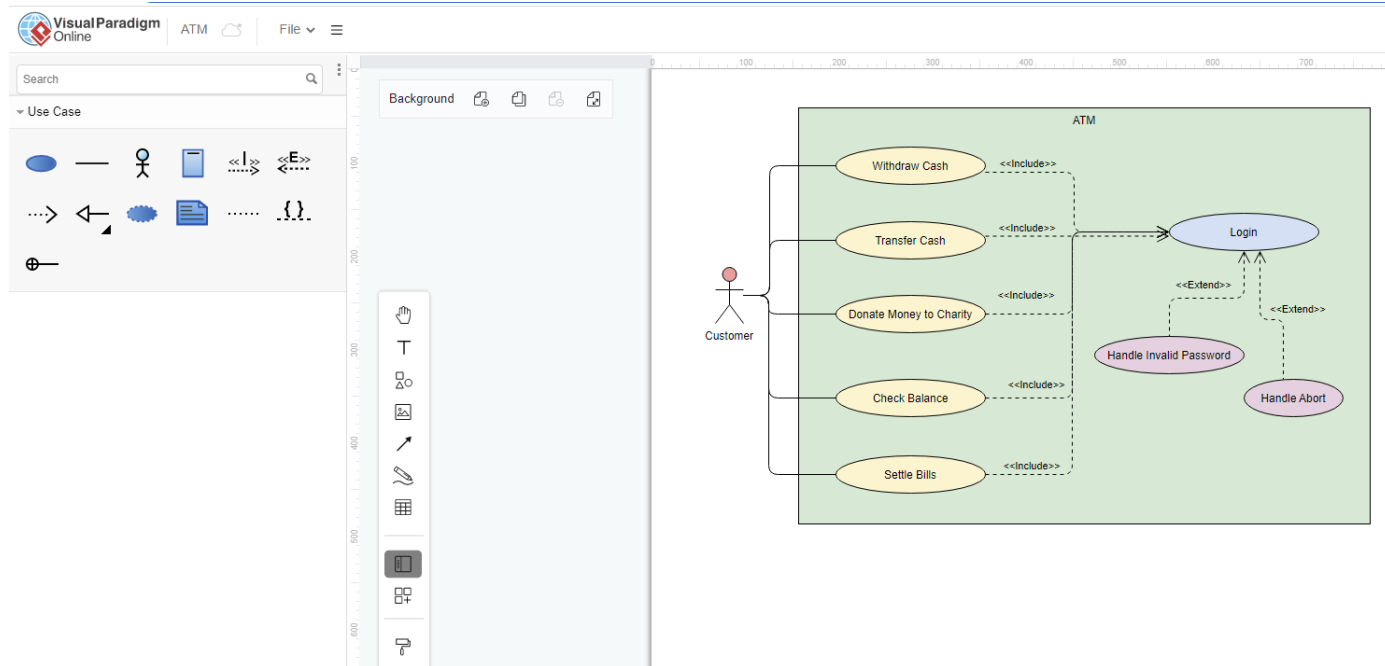
The third diagram shows an 'Online Ticketing System' UML Use Case Diagram. It features a central box labeled 'System' containing numerous green oval use cases. The use cases are arranged in a hierarchical structure: '1. Use Case' and '2. Use Case' are connected to a red stick figure actor on the left; '3. Use Case' and '4. Use Case' are connected to a red stick figure actor on the right; '5. Use Case' and '6. Use Case' are connected to a red stick figure actor on the left; and '7. Use Case' and '8. Use Case' are connected to a red stick figure actor on the right. The use cases are interconnected with lines, representing relationships between different functions of the system.

<https://creately.com/diagram-community/examples/t/use-case>

Программные среды для реализации UML-диаграмм



Программные среды для реализации UML-диаграмм



<https://online.visual-paradigm.com/diagrams/features/uml-tool/>