Центросоюз РФ

Нижегородский областной союз потребительских обществ

Нижегородский экономико-технологический колледж

|  |
| --- |
|  |

Цикловая комиссия естественнонаучных дисциплин и информационных

технологий

Специальность 09.02.07 Информационные системы и программирование

РЕФЕРАТ

Обзор диаграмм UML

|  |
| --- |
|  |

Выполнил студент группы 33П

Ананьев Артём Алексеевич

|  |
| --- |
|  |

СОДЕЖАНИЕ

Введение……………………………………………………………………….3

1. Назначение языка UML…………………………………………………....4
2. Виды диаграмм UML……………………………………………………....6
3. Структурные диаграммы………………………………………………….7
4. Диаграммы поведения……………………………………………………13
5. Диаграммы взаимодействия……………………………………………..18

Заключение…………………………………………………………………...19

Список литературы…………………………………………………………..21

Введение

UML - это унифицированный графический язык моделирования для описания, визуализации, проектирования и документирования ОО систем. UML призван поддерживать процесс моделирования ПС на основе ОО подхода, организовывать взаимосвязь концептуальных и программных понятий, отражать проблемы масштабирования сложных систем. Модели на UML используются на всех этапах жизненного цикла ПС, начиная с бизнес-анализа и заканчивая сопровождением системы. Разные организации могут применять UML по своему усмотрению в зависимости от своих проблемных областей и используемых технологий. К середине 90-х годов различными авторами было предложено несколько десятков методов ОО моделирования, каждый из которых использовал свою графическую нотацию. При этом любой их этих методов имел свои сильные стороны, но не позволял построить достаточно полную модель ПС, показать ее «со всех сторон», то есть, все необходимые проекции (См. статью 1). К тому же отсутствие стандарта ОО моделирования затрудняло для разработчиков выбор наиболее подходящего метода, что препятствовало широкому распространению ОО подхода к разработке ПС. По запросу Object Management Group (OMG) - организации, ответственной за принятие стандартов в области объектных технологий и баз данных назревшая проблема унификации и стандартизации была решена авторами трех наиболее популярных ОО методов - Г.Бучем, Д.Рамбо и А.Джекобсоном, которые объединенными усилиями создали версию UML 1.1, утвержденную OMG в 1997 году в качестве стандарта. Любой язык состоит из словаря и правил комбинирования слов для получения осмысленных конструкций. Так, в частности, устроены языки программирования, таковым является и UML. Отличительной его особенностью является то, что словарь языка образуют графические элементы. Каждому графическому символу соответствует конкретная семантика, поэтому модель, созданная одним разработчиком, может однозначно быть понята другим, а также программным средством, интерпретирующим UML. Отсюда, в частности, следует, что модель ПС, представленная на UML, может автоматически быть переведена на ОО язык программирования (такой, как Java, C++, VisualBasic), то есть, при наличии хорошего инструментального средства визуального моделирования, поддерживающего UML, построив модель, мы получим и заготовку программного кода, соответствующего этой модели.

Назначение языка UML

UML (от англ. Unified Modeling Language) переводится как «унифицированный язык моделирования». Это графический язык, в котором каждой фигуре, символу, стрелке или их сочетаниям присвоены конкретные значения. Он позволяет визуализировать явление или процесс так, чтобы схема была понятна всем, кто знаком с UML.

Можно сказать, что UML — это набор правил, по которым нужно рисовать схемы. Зная его, можно быстро создавать универсальные графические представления сложных процессов и структур. Именно поэтому IT-специалисты во время разработки ПО часто используют UML-моделирование и проектирование процессов. Язык UML нужен, чтобы описать и визуализировать какую-то абстрактную модель. На практике это может быть:  
  
- Создание модели объекта. Например, описание базы данных.

- Создание модели процессов. Например, последовательность выполнения запросов ПО, чтобы клиент получил ожидаемый результат.  
  
- Схему на языке UML можно составить по уже существующему объекту или процессу либо создать на этапе проектирования, чтобы разрабатывать объект или отлаживать процесс. Диаграммы UML применяют в проектировании, презентациях, описании или создании документации.

- Схемы можно рисовать и без языка, но у использования UML для построения диаграмм есть несколько преимуществ:  
  
- Стандартизация. Схема будет понятна любому, кто знает UML. Это всё равно что говорить на общеизвестном языке. Конечно, можно использовать и выдуманный — по жестам и интонациям будет примерно понятно, о чём речь. Но чтобы однозначно передать сложную информацию, лучше использовать язык, понятный всем.  
  
- Полнота. В языке уже предусмотрены конкретные обозначения для всех сущностей, необходимых в схеме. Не придётся что-то выдумывать в процессе, повторяться или вызывать непонимание у коллег.  
  
- Распространённость. UML используют в IT, менеджменте, инженерии и других сферах. Он понятен и привычен для многих специалистов всего мира.  
  
- Инструменты для автоматизации. Они позволяют автоматически генерировать схемы UML на основе кода. Например, с помощью Umbrello можно быстро создавать наглядные представления кода. Есть и обратные инструменты, которые меняют код в ответ на изменения UML-схемы — например, Microsoft Visual Studio 2010 Feature Pack 2. Таким образом, освоение UML позволяет разработчикам автоматизировать работу.

Все диаграммы UML можно поделить на структурные и поведенческие. Первые описывают структуру сложных объектов и систем, вторые иллюстрируют взаимодействие с системой и процесс её работы. Внутри эти типы делятся на виды UML-диаграмм.

Виды диаграмм UML

Структурные диаграммы - это тип диаграмм, используемых в различных областях, таких как программирование, системный анализ и проектирование. Они помогают визуализировать структуру системы, включая компоненты и их взаимосвязи.

Диаграммы поведения – это тип диаграмм, которые используются для моделирования динамических аспектов системы, таких как взаимодействие между объектами, последовательность действий и состояние системы. В UML (Unified Modeling Language) существует несколько типов диаграмм поведения.

Диаграммы взаимодействия в языке моделирования UML — это **диаграммы, на которых представлено взаимодействие множества объектов и отношений между ними, включая сообщения, которыми они обмениваются.**

Структурные диаграммы

Структурные диаграммы UML, как видно из названия, иллюстрируют структуру системы, включая ее классы, объекты, пакеты, компоненты и другие элементы, а также установленные между ними связи.

#### Диаграмма классов

Поскольку многие проекты опираются на объектно-ориентированное программирование, где от разработчика требуется описание типов возможных функций, диаграммы классов — самый популярный вариант UML. На [диаграмме классов](https://www.lucidchart.com/pages/uml-class-diagram) приводится статичная структура системы, включая классы, их атрибуты, поведение и взаимосвязи.



Рисунок 1 – Диаграмма классов UML

Класс обозначается прямоугольником из трех блоков, расположенных один над другим: в верхнем блоке обязательно указывается имя класса, а в остальных двух — дополнительная информация: атрибуты, операции и поведение.

#### Диаграмма компонентов

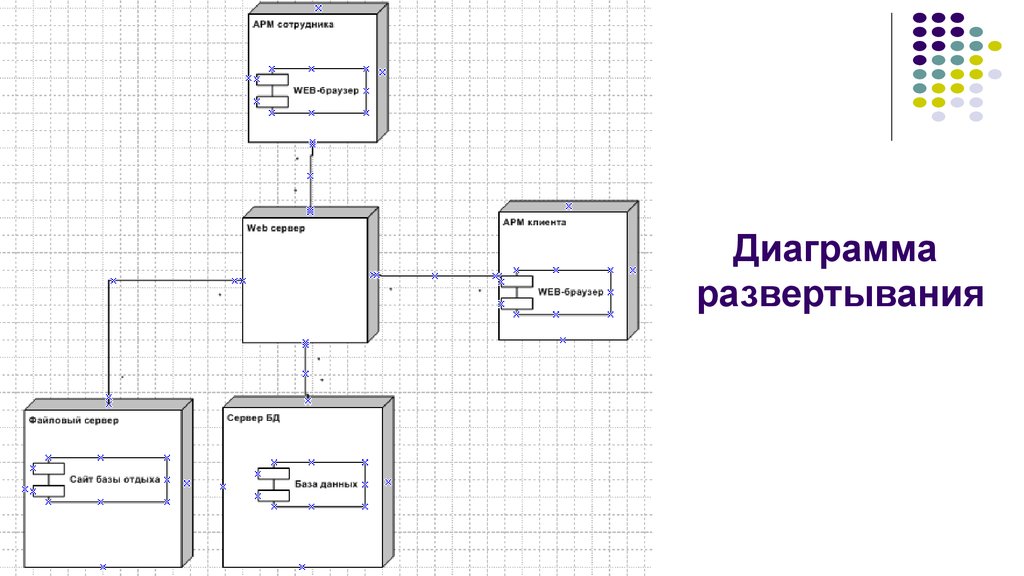
[Диаграмма компонентов](https://www.lucidchart.com/pages/uml-component-diagram) — по сути, более подробная версия диаграммы классов: и в той, и в другой действуют одни и те же правила. Диаграмма компонентов позволяет разбить комплексную систему на более мелкие составляющие и наглядно продемонстрировать установленные между ними связи.

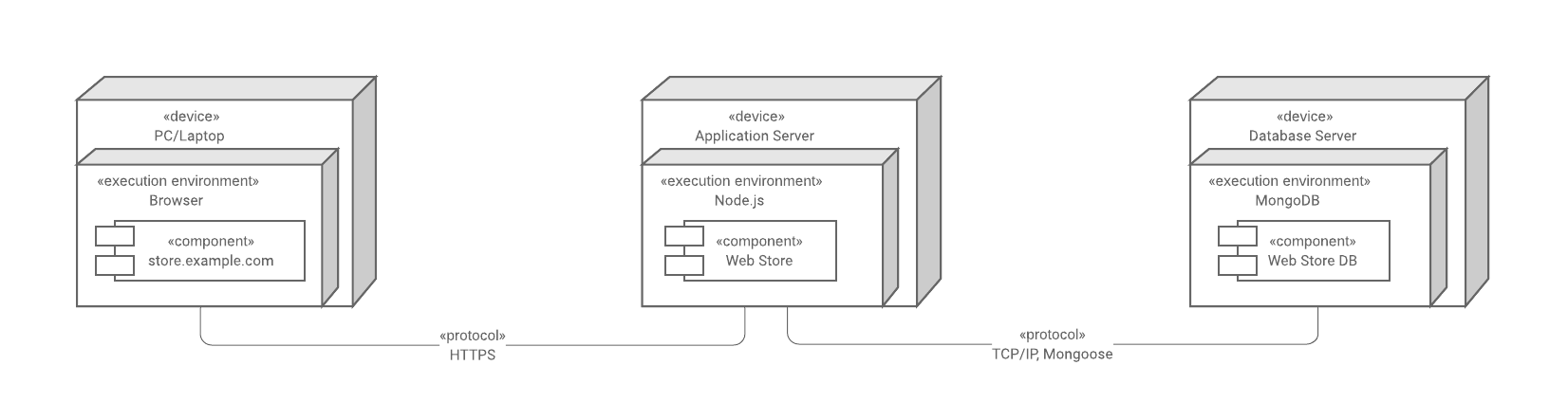


#### Рисунок 2 – Диаграмма компонентов UML

#### Диаграмма развертывания

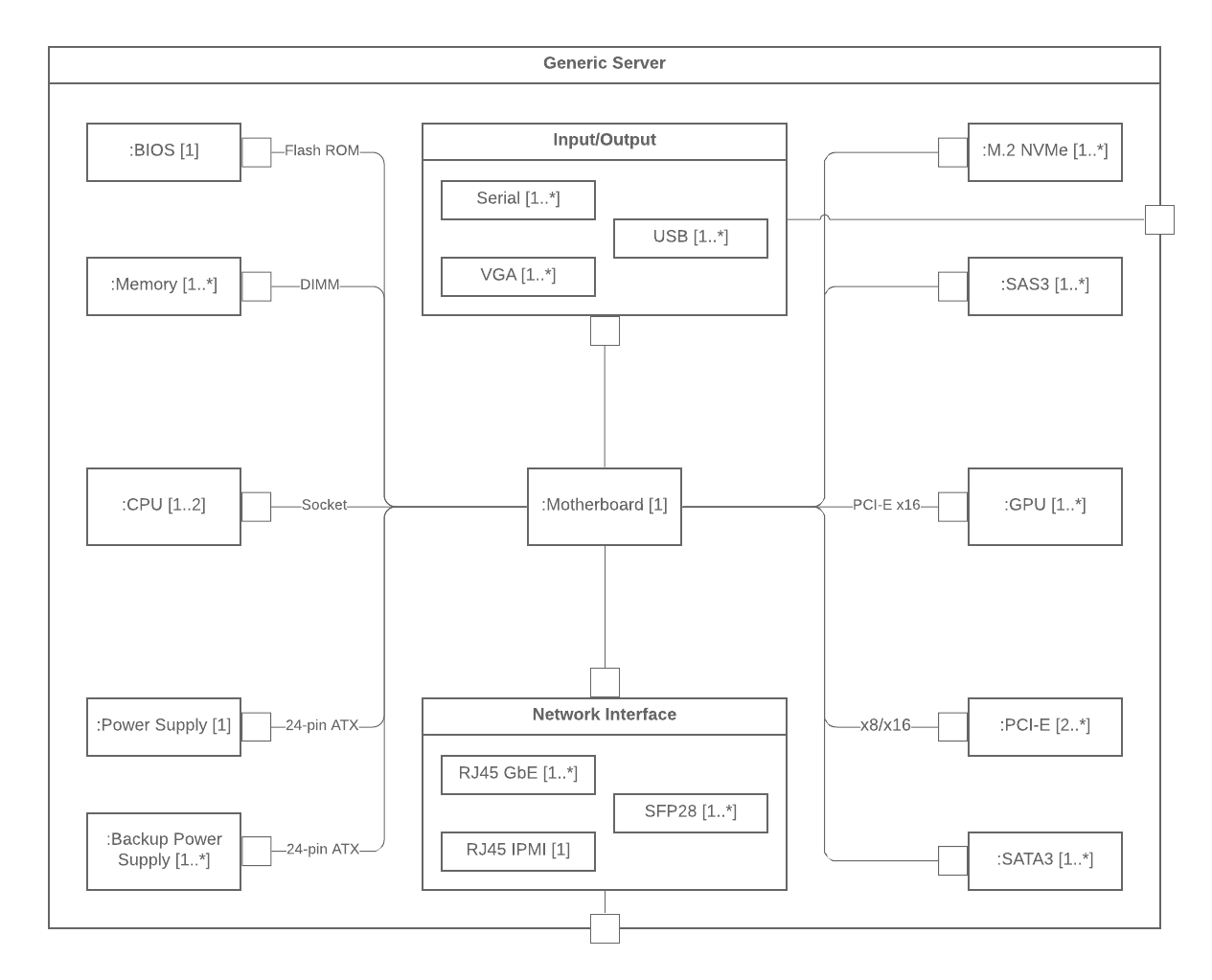
[Диаграммы развертывания](https://www.lucidchart.com/pages/ru/uml-deployment-diagram) иллюстрируют, как происходит развертывание программы на аппаратных компонентах системы. Чаще всего такими диаграммами пользуются проектировщики систем — для наглядной оценки производительности, масштабируемости, надежности и портативности. Поскольку на этих диаграммах показано, как аппаратные компоненты расположены относительно друг друга, с их помощью проще уследить за всей системой и не упустить из виду ни один компонент в ходе развертывания.



[](https://www.lucidchart.com/pages/ru/templates/uml-deployment-diagram-example)Рисунок 3 - Диаграмма развертывания UML

#### Диаграмма композитной структуры

Диаграммы этого типа наглядно описывают внутреннюю структуру классификатора. С их помощью также можно проиллюстрировать кооперацию или взаимодействие классификатора с собственной средой через порты. А еще диаграммы композитной структуры позволяют без труда обрисовать внутренние компоненты любого аппаратного обеспечения, чтобы ясно представить механизмы его работы.

[](https://www.lucidchart.com/pages/templates/uml-composite-structure-diagram)Рисунок 4 – диаграмма композитной структуры

#### Диаграмма объектов

[Диаграммы объектов](https://www.lucidchart.com/pages/uml-object-diagram) дают нам примеры того, как структуры данных выглядят в конкретный момент времени. Если статичную структуру удобно представить в виде диаграммы классов, то диаграммы объектов, как контрольные примеры, позволяют судить, все ли ее компоненты на месте. Из диаграммы объектов также можно почерпнуть полезную информацию об элементах модели и их ссылках.



#### Рисунок 5 – Диаграмма объектов UML

#### Диаграмма пакетов

Диаграммы пакетов наглядно демонстрируют зависимости между пакетами в составе системы. Пакет изображается в виде папки и применяется для организации элементов модели (например, классов или сценариев использования) в группы.



Рисунок 6 – Диаграмма пакетов UML

Диаграммы поведения

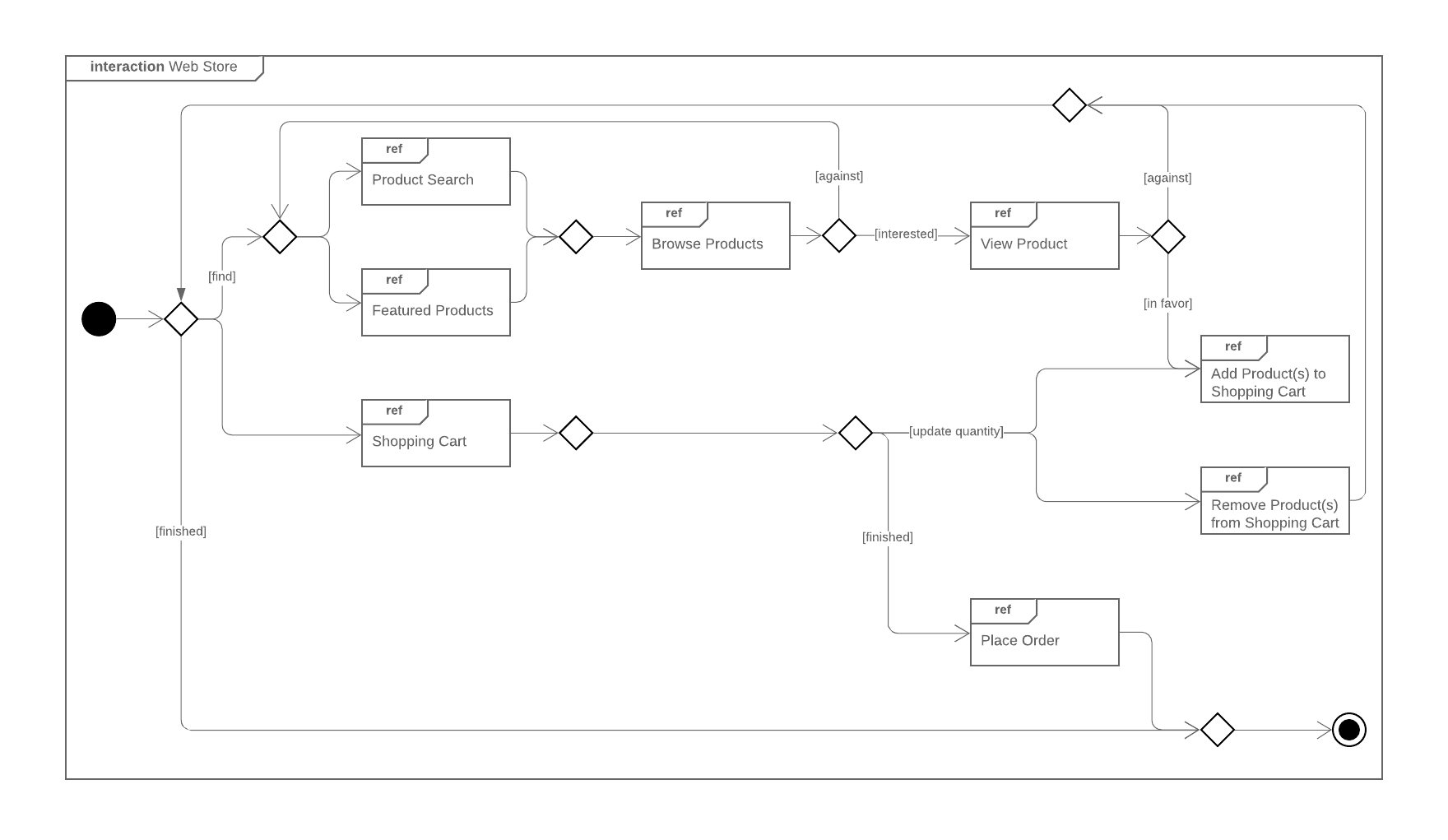
Диаграммы этой категории иллюстрируют поведение системы и ее взаимодействие с пользователями, другими системами и прочими сущностями.

#### Временная диаграмма

[Временные диаграммы](https://www.lucidchart.com/pages/uml-timing-diagram) часто называют «диаграммами последовательностей наоборот», а их задача заключается в том, чтобы показать, как объекты взаимодействуют между собой в определенный период времени. Эти диаграммы позволяют оценить, сколько времени занимает каждый этап процесса, и выявить участки для доработки.

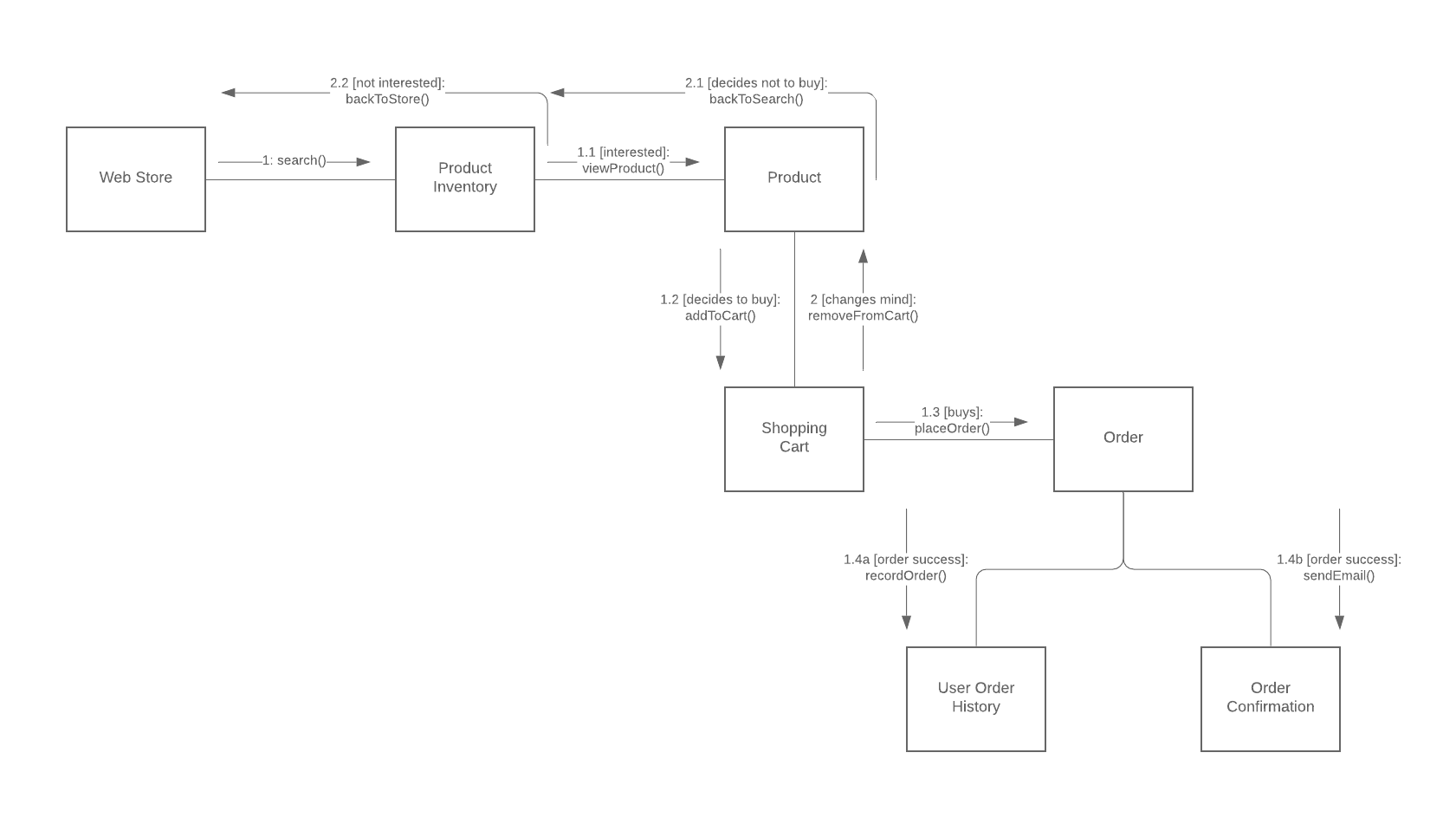
#### Диаграмма обзора взаимодействия

Эти диаграммы наглядно описывают движение потока управления между взаимодействующими узлами. К последним относятся начальные узлы, конечные узлы потока или деятельности, а также узлы решений, слияния, разделения и соединения.

[](https://www.lucidchart.com/pages/templates/uml-interaction-overview-diagram)Рисунок 7 -диаграмма обзора взаимодействия UML

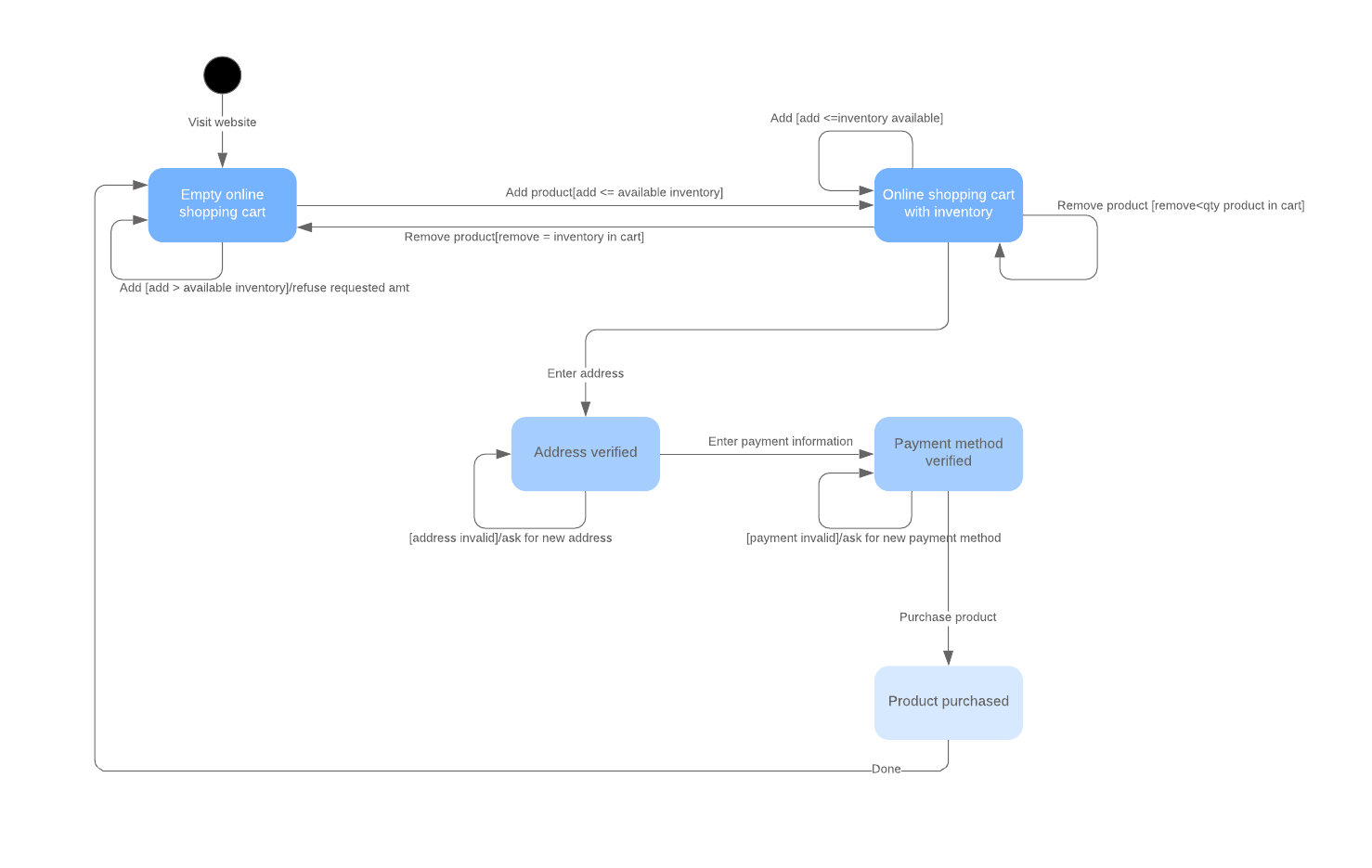
#### Диаграмма коммуникации

[Диаграммы коммуникации](https://www.lucidchart.com/pages/uml-communication-diagram) (раньше они назывались диаграммами кооперации) показывают связи между объектами. По этим диаграммам можно судить, как объекты взаимодействуют друг с другом посредством сообщений в пределах архитектуры системы. Здесь же можно рассмотреть альтернативные варианты в рамках сценариев использования или операций, где требуется взаимодействие разных объектов.

[](https://www.lucidchart.com/pages/templates/uml-communication-diagram)Рисунок 8 - диаграмма коммуникации UML

#### Диаграмма состояний

[Диаграммы состояния](https://www.lucidchart.com/pages/uml-state-machine-diagram), если объяснять в двух словах, иллюстрируют состояния и переходы. Под «состоянием» подразумевается то или иное сочетание данных, которое способен содержать объект, а на самой диаграмме можно показать не только все возможные состояния, но и способы перехода объекта из одного состояния в другое.

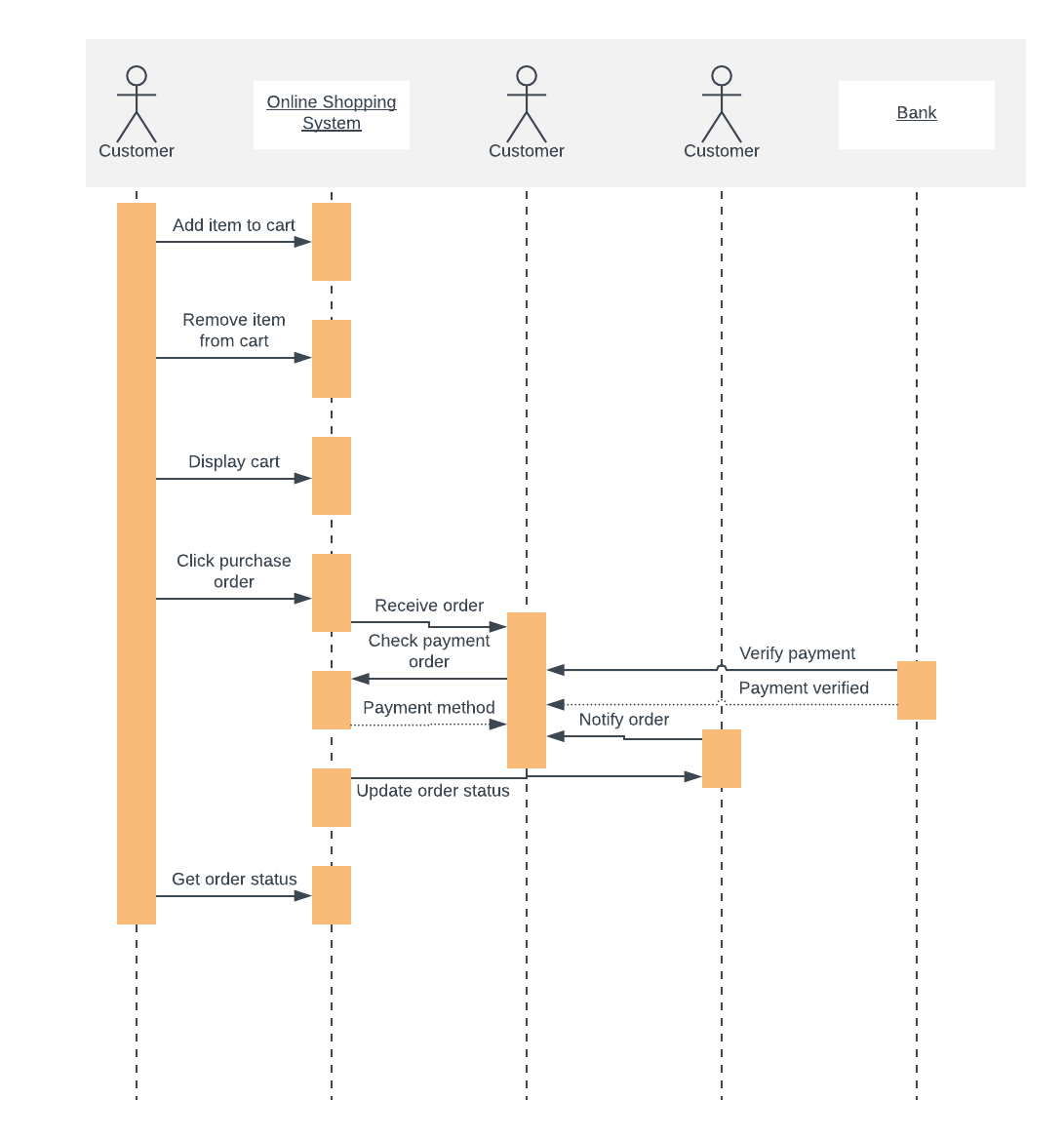
[](https://www.lucidchart.com/pages/ru/templates/uml-state-diagram)Рисунок 9 - диаграмма состояний UML

#### Схема сценариев использования

[Диаграммы сценариев использования](https://www.lucidchart.com/pages/uml-use-case-diagram) показывают, как пользователи (которые называются «агентами» и обозначаются в виде человечков-палочек) взаимодействуют с системой. Диаграммы этого типа служат обобщенной моделью связей между агентами и системами и помогут вам объяснить устройство системы даже аудитории без особой технической подготовки.

#### Диаграмма последовательностей

[Диаграммы последовательностей](https://www.lucidchart.com/pages/uml-sequence-diagram), которые иногда также называют диаграммами (или сценариями) событий, показывают, в каком порядке происходит взаимодействие между объектами. С помощью таких диаграмм довольно легко составлять незамысловатые сценарии рабочих циклов.

[](https://www.lucidchart.com/pages/templates/sequence-diagram-for-online-shopping)Рисунок 10 - диаграмма последовательностей UML

#### Диаграмма активности

На [диаграммах активности](https://www.lucidchart.com/pages/uml-activity-diagram) можно показать разные этапы реализации сценария использования. Действия (или «активности») могут следовать одно за другим, ответвляться друг от друга либо разворачиваться одновременно. Таким образом, система предстает перед нами в динамике, а диаграммы активности также могут пригодиться в [моделировании бизнес-процессов](https://www.lucidchart.com/blog/business-process-modeling-methodologies).

Если вам не хватает вдохновения, предлагаем вашему вниманию полную статью с [примерами и шаблонами диаграмм UML](https://www.lucidchart.com/blog/uml-diagram-templates).

Эти шаблоны послужат отличной отправной точкой для работы, а также лишний раз докажут основной тезис нашей публикации: диаграммы UML должны быть не факультативным дополнением, а обязательной составляющей документации для любого проекта в гибкой среде.

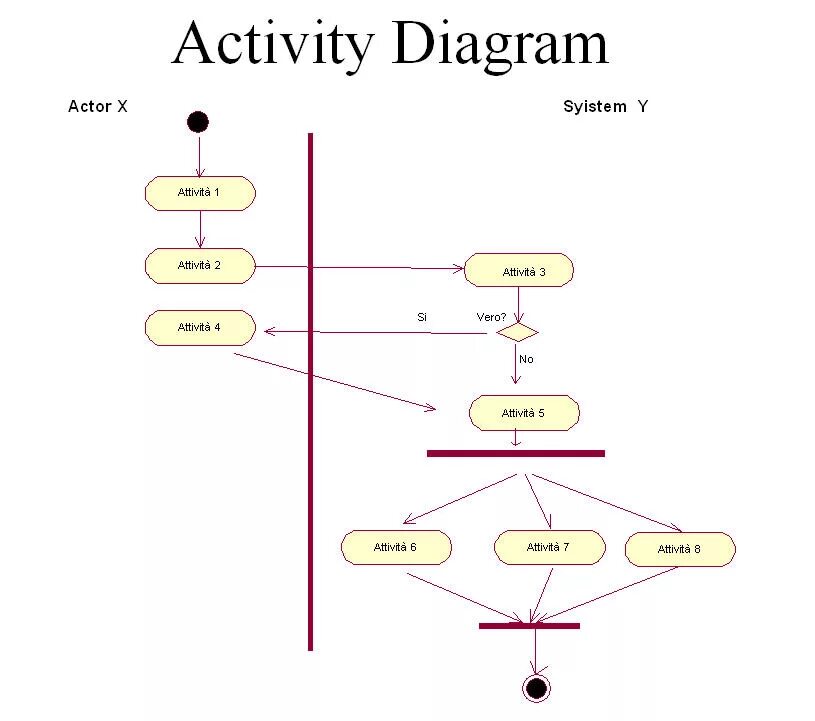


Рисунок 11 – Диаграмма активности UML

Диаграммы взаимодействия

Диаграмма взаимодействия одна из разновидностей диаграммы деятельности в языке моделирования UML, которая может включать в себя также элементы диаграммы последовательности. Цель её создания ставится как увязывание в единое целое потока управления между узлами из диаграмм деятельности с последовательностью сообщений между линиями выполнения диаграмм последовательности. Расширение синтаксиса осуществляется за счёт использования ссылок на взаимодействия, которые основаны на диаграмме последовательности.

Основным назначением диаграммы обзора взаимодействий является создание высокоуровневого представления об общем характере взаимодействий в проектируемой системе. Она не концентрирует своё внимание на отдельных аспектах работы так, как это происходит в других видах поведенческих диаграмм (коммуникации, последовательности, взаимодействия и т.п.) При её конструировании она компонуется с использованием нотации диаграммы последовательности, однако обозначения ветвлений и разделений заимствуются из диаграммы деятельности.

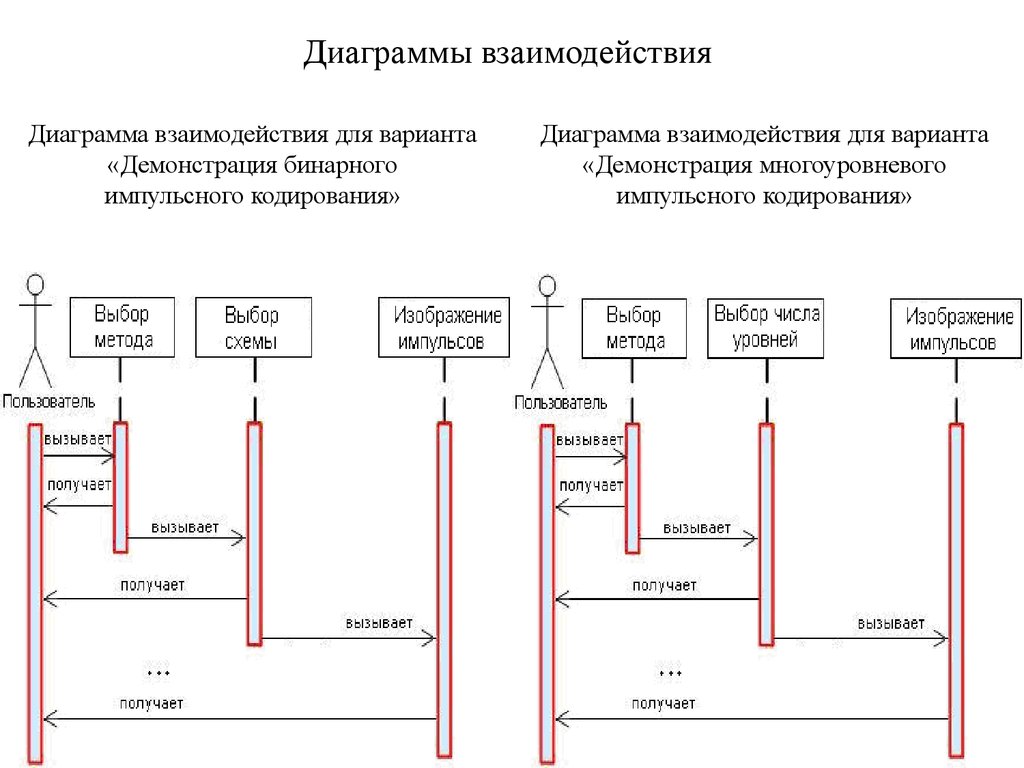


Рисунок 12 – Диаграмма взаимодействия UML

Заключение

**Унифицированный язык моделирования (Unified Modeling Language, UML)** — это графический язык для спецификации, визуализации, проектирования и документирования систем. С помощью UML можно разработать подробную модель создаваемой системы, отображающую не только её концепцию, но и конкретные особенности реализации. В рамках UML-модели все представления о системе фиксируются в виде специальных графических конструкций, получивших название диаграмм. Слово "моделирование", входящее в название UML, имеет множество смысловых оттенков и сложившихся способов употребления. В частности, английские слова modeling и simulation оба переводятся словом "моделирование", хотя означают разные вещи. В первом случае речь идет о составлении модели, которая используется только для описания моделируемого объекта или явления. Во втором случае подразумевается составление модели, которая может быть использована для получения существенной информации о моделируемом объекте или явлении. При этом во втором случае обычно добавляется уточняющее прилагательное: численное моделирование, математическое моделирование и др. UML является языком моделирования в первом смысле, хотя известны некоторые успешные попытки использования UML и во втором смысле. В отношении разработки программного обеспечения так сложилось, что результаты фаз анализа и проектирования, оформленные средствами определенного языка, принято называть моделью∇. Деятельность по составлению моделей естественно назвать моделированием. Именно в этом смысле UML является языком моделирования. В этой книге термины "программное обеспечение", "программная система", "программа" и "приложение" используются как синонимы. Авторы понимают, что в действительности между ними существуют определенные различия, однако в контексте книги эти различия не имеют большого значения. Таким образом, модель UML ‒ это, прежде всего, описание объекта или явления, а также и кое-что другое, а именно все, что авторам UML удалось включить в язык, не нарушая принципа унификации, к изложению которого мы переходим в следующем разделе. Описывая историю создания UML, его авторы характеризуют эпоху до UML как период "войны методов". Пожалуй, "война" ‒ это слишком сильно сказано, но, действительно, UML является отнюдь не первым языком моделирования. К моменту его появления насчитывались десятки других, различающихся системой обозначений, степенью универсальности, способами применения и т.д. Авторы языков и теоретики программирования препирались между собой, выясняя, чей подход лучше, а разработчики всю эту "войну методов" равнодушно игнорировали, поскольку ни один из методов не дотягивал до уровня индустриального стандарта. Приложив заслуживающие уважения усилия, авторы UML при поддержке и содействии всей международной программистской общественности смогли свести воедино (унифицировать) большую часть того, что было известно и до них. В результате унификации получилась теоретически изящная и практически полезная вещь ‒ UML.

Список литературы

### <https://profspo.ru/books/83601> - Проектирование информационных систем по методологии UML с использованием Qt-технологии программирования (Галиаскаров Э. Г., Воробьев А. С.)

### <https://urait.ru/book/analiz-i-proektirovanie-sistem-s-ispolzovaniem-uml-544559> - Анализ и проектирование систем с использованием UML ([Дерябкин В.П., Козлов В.В.](https://profspo.ru/catalog/?authors=%D0%94%D0%B5%D1%80%D1%8F%D0%B1%D0%BA%D0%B8%D0%BD%20%D0%92.%D0%9F.,%20%D0%9A%D0%BE%D0%B7%D0%BB%D0%BE%D0%B2%20%D0%92.%D0%92.))

### <https://urait.ru/book/programmnaya-inzheneriya-vizualnoe-modelirovanie-programmnyh-sistem-539955> - Программная инженерия. Визуальное моделирование программных систем(Черткова Е. А.)

### <https://ru.wikipedia.org/wiki/UML> - Wikipedia

### <https://skillbox.ru/media/code/yazyk-uml-chto-eto-takoe-i-zachem-on-nuzhen/?ysclid=m2lpuknhou157152358> – Skillbox

### <https://practicum.yandex.ru/blog/uml-diagrammy/> - Яндекс Практикум

### <https://www.bibliofond.ru/view.aspx?id=792249> – Библиофонд

### <https://www.lucidchart.com/blog/types-of-UML-diagrams> - Lucidchart