*Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ \_\_ОЭП\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_РЛ2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Реферат**

**на тему:**

**Объектно-ориентированное моделирование бизнес-процессов**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** \_\_\_Фильчаков А.И.\_\_

(Подпись, дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_**Прокудин В.Н.**\_**

(Подпись, дата) (И.О.Фамилия)

Москва, 2020

Содержание

[Введение 3](#_Toc41399970)

[1. Основные принципы и понятия ООМ 4](#_Toc41399971)

[2. Иерархия структуры 4](#_Toc41399972)

[3. Диаграммы UML 7](#_Toc41399973)

[4. Описание работы транспортной системы 18](#_Toc41399974)

[Заключение 22](#_Toc41399975)

[Список используемой литературы 23](#_Toc41399976)

**Введение**

Рыночная экономика требует от предприятия торговли повышение эффективности деятельности, конкурентоспособности реализуемых товаров и услуг на основе внедрения достижений научно-технического прогресса, эффективных форм хозяйствования, квалифицированного маркетинга и управления торговой деятельностью, преодоление бесхозяйственности, активизации предпринимательства, инициативы и т. д.

Всякая деятельность любой коммерческой и некоммерческой фирмы включают в себя большое количество повторяющихся бизнес-процессов. Каждый из них является совокупностью последовательных действий и решение, которые направлены на достижение поставленных целей. Успешность фирмы на рынке, конкурентоспособность, прибыльность определяется степенью автоматизации и эффективностью бизнес-процессов, протекающих в фирме.

1. Основные принципы и понятия ООМ

Понятие объектно-ориентированного моделирования (ООМ), безусловно, связано с объектно-ориентированным программированием (ООП). Этот подход к разработке программных средств, появившийся в средине 1980-х гг., первоначально был направлен на разрешение проблем, возникающих в результате неизбежного роста и усложнения программ, а также задач обработки данных и манипулирования ими. В то время стало очевидным, что традиционные методы процедурного программирования нe способны справиться ни с растущей сложностью программ и их разработки, ни с необходимостью повышения их надежности. При этом вычислительные и расчетно-алгоритмические задачи, особенно в области обеспечения бизнеса, постепенно стали уходить на второй план, а первое место стали занимать задачи именно обработки данных и манипулирования ими.

Фундаментальными понятиями ООП являются понятия класса и объекта. При этом под классом понимается некоторая абстракция совокупности объектов, которые имеют общий набор свойств и обладают одинаковым поведением. Каждый объект в этом случае рассматривается как экземпляр соответствующего класса. Объекты, которые не имеют полностью одинаковых свойств или не обладают одинаковым поведением, по определению, не могут быть отнесены к одному классу. Хотя приведенное определение класса может быть уточнено на основе учета других понятий ООП, оно является общим и достаточным для проведения ООМ [1].

# Иерархия структуры

Важная особенность классов состоит в возможности их организации в виде некоторой иерархической структуры, которая по внешнему виду напоминает схему классификации понятий формальной логики. В связи с этим следует отметить, что каждое понятие в логике имеет объем и содержание. Под объемом понятия понимают все другие мыслимые понятия, для которых исходное понятие может служить определяющей категорией или частью. Содержание же понятия составляет совокупность всех его признаков и атрибутов, отличающих данное понятие от всех других. В формальной логике известен закон обратного отношения: если содержание понятия А содержится в содержании понятия В, то объем понятия В содержится в объеме понятия А.

Иерархия понятий строится следующим образом. В качестве наиболее общего понятия или категории берется понятие, имеющее наибольший объем и, соответственно, наименьшее содержание. Это самый высокий уровень абстракции для данной иерархии. Затем общее понятие некоторым образом конкретизируется, тем самым уменьшается его объем и увеличивается содержание. Появляется менее общее понятие, которое на схеме иерархии будет расположено на уровень ниже исходного понятия. Этот процесс конкретизации понятий может быть продолжен до тех пор, пока на самом нижнем уровне не будет получено понятие, дальнейшая конкретизация которого невозможна либо нецелесообразна.

Очевидно, что если в качестве самого абстрактного понятия для достижения целей ООМ принять некоторую модель, то концептуальную схему объектно-ориентированного моделирования можно представить, как показано на рисунке 1.

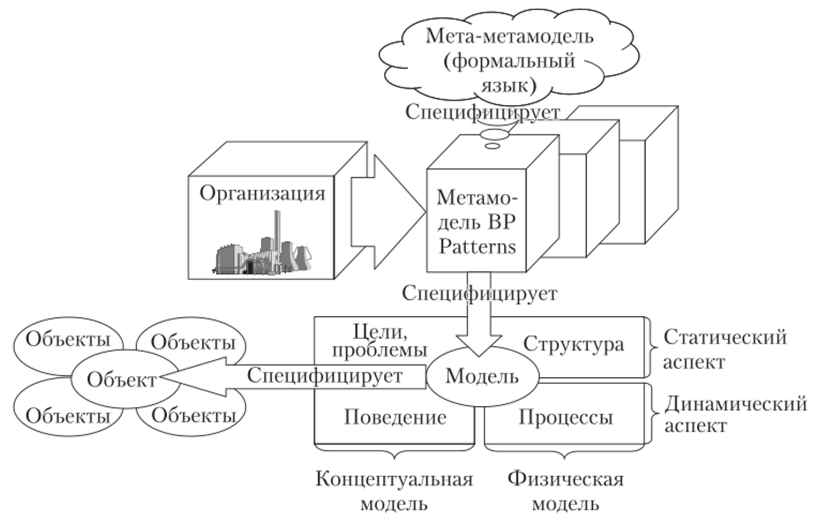


Рисунок 1 – Концептуальная схема ООМ

В настоящее время языком, реализующим объектно-ориентированные подходы (в том числе и к моделированию бизнес-процессов), является язык UML (Unified Modeling Language), представляющий собой общецелевой язык визуального моделирования, который разработан для спецификации, визуализации, проектирования и документирования компонентов программного обеспечения, бизнес-процессов и других систем. Этот язык может быть использован для построения концептуальных, логических и графических моделей сложных систем различного целевого назначения.

Формальное описание предметной области с использованием UML основывается на иерархической структуре модельных представлений (см. рисунок 1), состоящей из четырех уровней: 1) мета-метамодели; 2) метамодели; 3) модели; 4) объектов.

Уровень мета-метамодели образует исходную основу для всех мета- модельных представлений. Главное его назначение состоит в том, чтобы определить язык для спецификации метамодели. Мета-метамодель определяет формальный язык на самом высоком уровне абстракции и является наиболее компактным его описанием. При этом мета-метамодель может специфицировать несколько метамоделей, чем достигается потенциальная гибкость включения дополнительных понятий.

Метамодель является экземпляром или конкретизацией мета-мета- модели. Главная задача этого уровня — определить язык для спецификации модели. Данный уровень более конструктивный, чем предыдущий, поскольку обладает более развитой семантикой базовых понятий.

Модель в контексте языка UML — это экземпляр метамодели в том смысле, что любая конкретная модель системы должна использовать только понятия метамодели, конкретизировав их применительно к данной ситуации. Это уровень для описания информации о конкретной предметной области. Однако если для построения модели используются понятия языка UML, то необходима полная согласованность понятий уровня модели с понятиями языка уровня метамодели. Конкретизация же понятий модели происходит на уровне объектов.

# ДиаграммыUML

С самой общей точки зрения UML состоит из двух взаимодействующих частей: семантики языка и нотации. Семантика определена для двух видов моделей: структурных моделей и моделей поведения. Структурные модели, известные также как статические, описывают структуру сущностей или компонентов (элементов) некоторой системы, включая их атрибуты и отношения. Модели поведения, называемые иногда динамическими моделями, описывают поведение или функционирование объектов системы, взаимодействие между ними, а также процесс изменения состояний отдельных элементов и системы в целом. Следует отметить, что именно для отображения поведенческого аспекта систем, в первую очередь, и создавался UML. Нотация же языка представляет собой графическую спецификацию для визуального представления семантики языка.

В рамках языка UML все представления о модели сложной системы фиксируются в виде специальных графических конструкций — диаграмм. В терминах UML определены следующие виды диаграмм (см. рисунок 2):

• диаграмма вариантов использования (Use Case Diagram);

• диаграмма классов (Class Diagram);

• диаграммы поведения (Behavior Diagrams);

• диаграммы реализации (Implementation Diagrams).

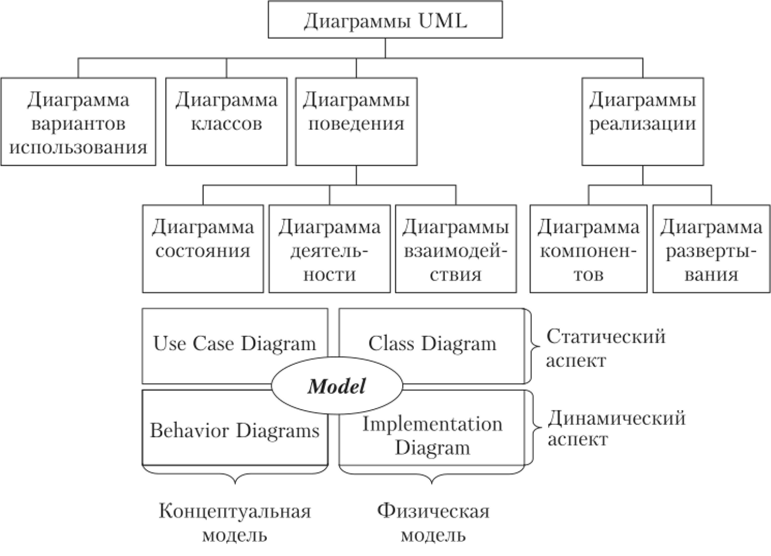


Рисунок 2 – Диаграммы UML (уровень метамодели)

Каждая из этих диаграмм детализирует и конкретизирует различные представления о модели сложной системы в терминах языка UML. При этом диаграмма вариантов использования представляет собой наиболее общую концептуальную модель сложной системы, которая является исходной для построения других диаграмм. Диаграмма классов является по своей сути логической моделью, отражающей статические аспекты структурного построения системы. [2]

Особую роль играют диаграммы поведения, призванные отражать динамические аспекты функционирования сложной системы. К этому виду диаграмм относятся:

* диаграмма состояния (Statechart Diagram);
* диаграмма деятельности (Activity Diagram);
* диаграммы взаимодействия (Interaction Diagrams):
* диаграмма последовательности (Sequence Diagram);
* диаграмма кооперации (Collaboration Diagram).

Наконец, диаграммы реализации служат для представления физических компонентов сложной системы. К ним относятся:

* диаграмма компонентов (Component Diagram);
* диаграмма развертывания (Deployment Diagram).

В современной литературе довольно подробно рассмотрены все перечисленные диаграммы и объекты уровня метамодели.

С точки зрения моделирования бизнес-процессов визуальное моделирование в UML можно представить как некоторый процесс поуровневого спуска от наиболее общей и абстрактной концептуальной модели исходной системы к логической, а затем и к физической модели соответствующей программной системы. Для достижения этих целей сначала строится модель в форме так называемой диаграммы вариантов использования (Use Case Diagram), которая описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки.

Разработка диаграммы вариантов использования преследует следующие цели:

* определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы;
* сформулировать общие требования к функциональному поведению проектируемой системы;
* разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
* подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или акторов, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актором (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая способна служить источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, вариант использования (use case) служит для описания сервисов, которые система предоставляет актору. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актором. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие акторов с системой.

Основные объекты диаграммы вариантов использования сведены в таблице 1.

Таблица 1 – Основные объекты диаграммы вариантов использования UML

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Объект | Обозначение | | | Назначение | | |
| Вариант использования | F Проверить состояние (текущего счета) клиента банка | | | Вариант использования применяется для спецификации общих особенностей поведения системы или любой другой сущности предметной области без рассмотрения внутренней структуры этой сущности | | |
| Актор | * 0 * 1 | | | Актор представляет собой любую внешнюю, но отношению к моделируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей или решения частных задач | | |
| Интерфейс | о о  ДатчикУстройство считывания штрихкода | | | Интерфейс служит для спецификации параметров модели, которые видимы извне без указания их внутренней структуры | | |
| Примечание |  | Реализовать в виде отдельной библиотеки стандартных функций  I |  | | Примечания в языке UML предназначены для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта |

|  |  |  |  |
| --- | --- | --- | --- |
| Отношения на диаграмме вариантов использования | | | |
| Отношение ассоциации (association relationship) |  | Ассоциация специфицирует семантические особенности взаимодействия акторов и вариантов использования в графической модели системы |
| Отношение расширения (extend relationship) |  | Отношение расширения определяет взаимосвязь экземпляров отдельного варианта использования с более общим вариантом, свойства которого определяются на основе способа совместного объединения данных экземпляров |
| Отношение обобщения (generalization relationship) |  | Отношение обобщения служит для указания того факта, что некоторый вариант использования Л может быть обобщен до варианта использования В. В этом случае вариант А будет являться специализацией варианта В |
| Отношение включения (include relationship) |  | Отношение включения между двумя вариантами использования указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования |

Пример диаграммы вариантов использования показан на рисунке 3.



Рисунок 3 – Диаграмма вариантов использования для системы продаж

С точки зрения имитационного моделирования наибольший интерес представляют диаграмма вариантов использования и диаграммы поведения. Именно эти диаграммы призваны описывать функциональность (активность, движение) компонентов системы. Кроме того, совокупность этих диаграмм полностью определяет концептуальный уровень описания сложной системы (концептуальную модель). В связи с этим следует рассмотреть указанные диаграммы более подробно. Для примера в качестве сложной системы примем отдел продаж и маркетинга некоторой фирмы. Основная цель моделирования данной системы заключается в автоматизации работы отдела, т.е. в создании и внедрении информационной системы управления продажами. При этом предполагается, что в отделе полностью отсутствует автоматизация производственной деятельности. [3]

Не вдаваясь в описание семантики языка UML (она хорошо освещена в соответствующей литературе), приведем лишь результаты объектно-ориентированного анализа, показанные на рисунках 4-8.

Нетрудно заметить, что время как важнейший атрибут любой поведенческой модели присутствует на приведенных диаграммах лишь опосредованно. Это означает, что при анализе поведения (или изменения состояний) возможны лишь качественные оценки типа «не раньше, чем...», «только после того, как...» и т.п. Однако при анализе, например, диаграммы состояний (см. рисунок 4) естественным образом возникают следующие вопросы: «Как часто поступают заказы?», «Как долго они оформляются?», «Каково соотношение количества автоматизированных рабочих мест (АРМ) и числа менеджеров?», «Какой должна быть производительность сервера?» и т.д. Очевидно, что без привлечения аппарата имитационного моделирования получить ответы на эти вопросы по приведенным диаграммам просто невозможно.

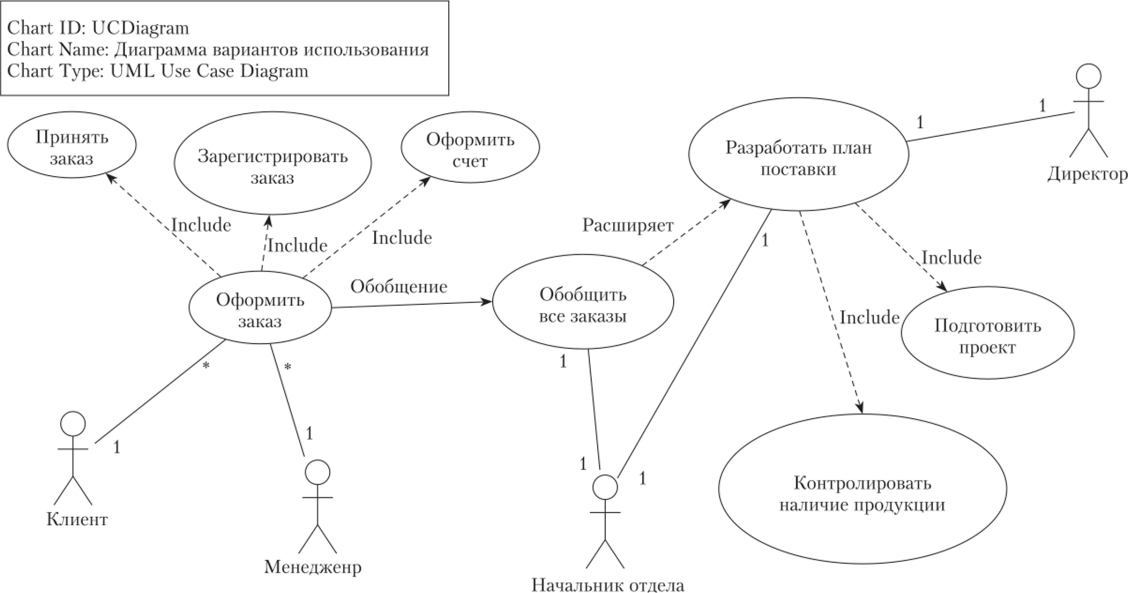


Рисунок 4 – Диаграмма вариантов использования

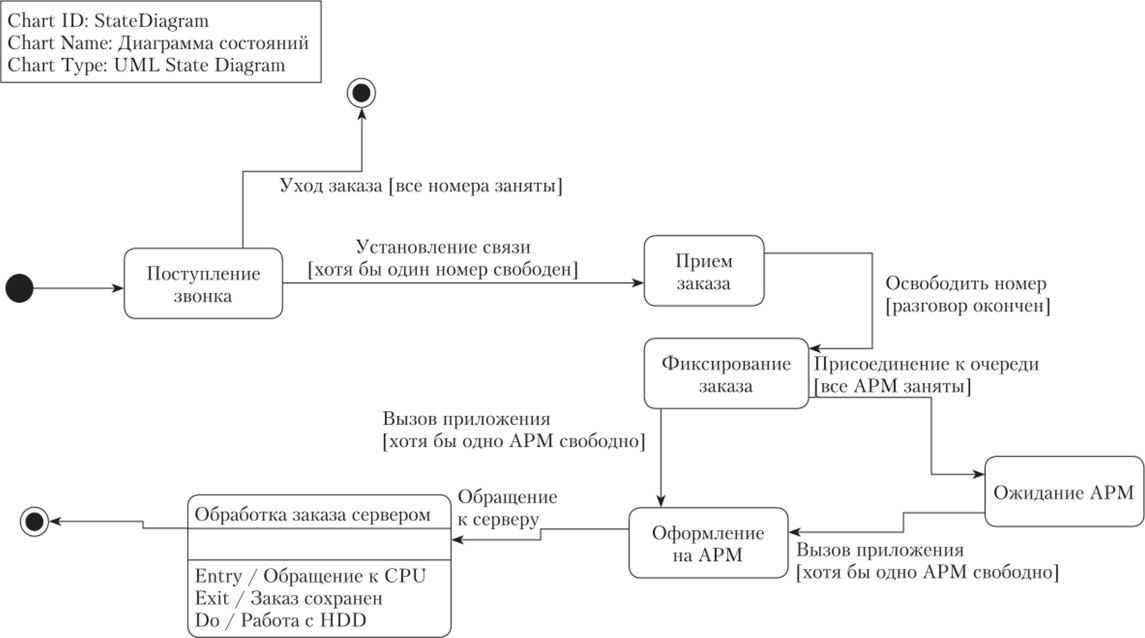


Рисунок 5 – Диаграмма состояний

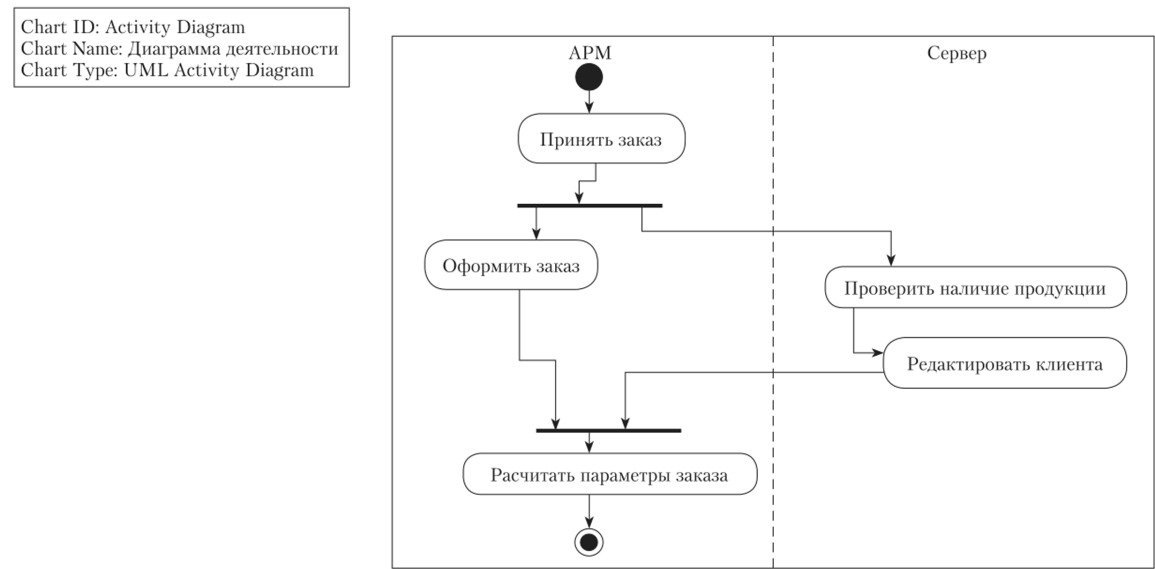


Рисунок 6 – Диаграмма деятельности

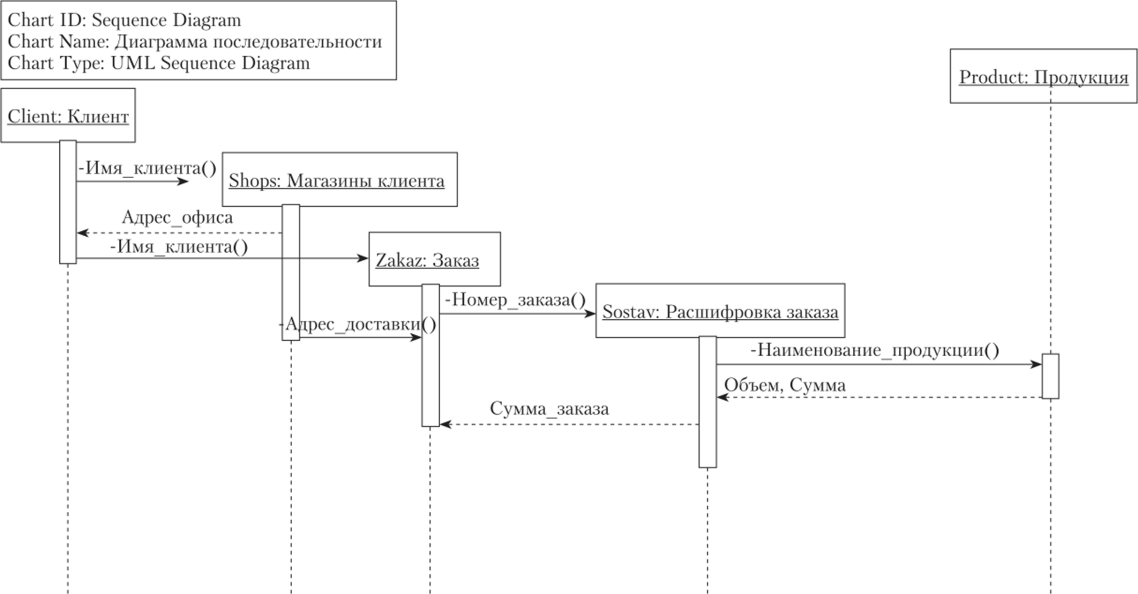


Рисунок 7 – Диаграмма деятельности

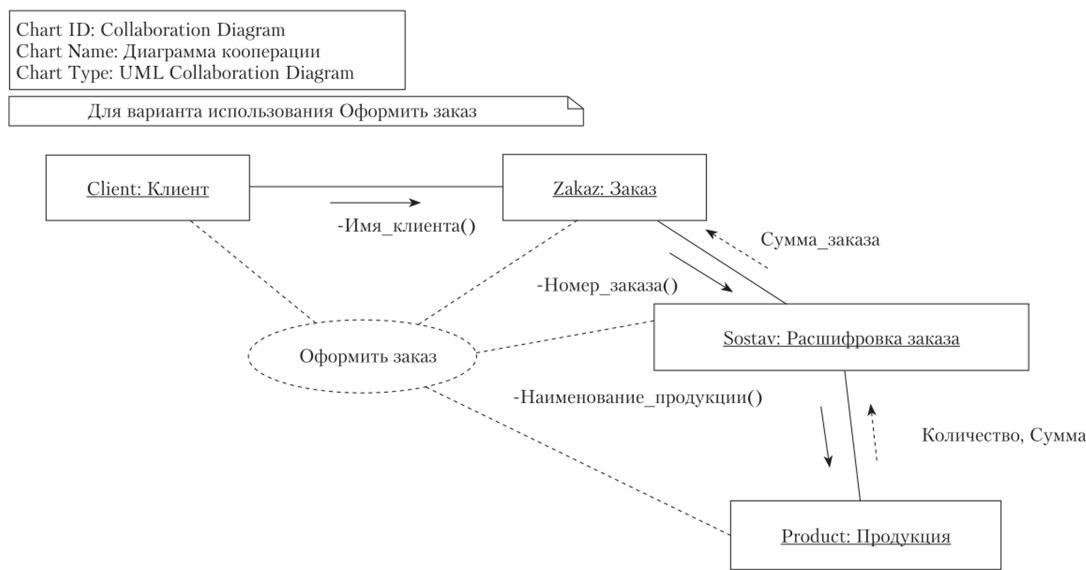


Рисунок 8 – Диаграмма кооперации

При этом отметим, что при построении заключительных диаграмм (диаграммы компонентов и диаграммы развертывания) требуется явно указывать технические характеристики аппаратных средств, например, таких, как количество рабочих станций (АРМ), тактовая частота процессоров, скорость передачи по сети, емкость запоминающих устройств и др. Ясно, что завышенные показатели могут привести к неоправданным затратам, а заниженные — к снижению эффективности функционирования всей системы. Поэтому обоснование требуемых значений всех технических показателей возможно лишь, но результатам имитационного моделирования.

Объектам диаграмм UML, моделирующим поведение системы, могут быть поставлены в соответствие объекты имитационной модели. Важнейшими диаграммами, несущими необходимую информацию, в данном случае являются диаграмма вариантов использования и диаграмма состояний. Взаимосвязь объектов этих диаграмм с объектами имитационной модели показана в таблице 2.

Таблица 2 – Взаимосвязь объектов диаграмм UML и объектов имитационной модели

|  |  |  |
| --- | --- | --- |
| Объект диаграммы вариантов использования | Объект диаграммы состояний | Объект имитационной модели |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Объект диаграммы вариантов использования | Объект диаграммы состояний | Объект имитационной модели |
|  |  |  |
|  |  |  |
|  |  |
|  |  |
|  |
|  |  |
|  |  |

# Описание работы транспортной системы

Для примера рассмотрим работу транспортной системы.Пример фрагмента иерархии классов для транспортной системы приведен на рисунке 9. При моделировании транспортной системы важной особенностью является отнесение объекта к активным или пассивным. Активные объекты могут самостоятельно генерировать определенный набор событий, например прибытие транспортного средства на терминал. Пассивные объекты в состоянии реагировать только на события, генерируемые извне. Пример разделения объектов транспортной системы на эти две группы приведен на рисунке 10.

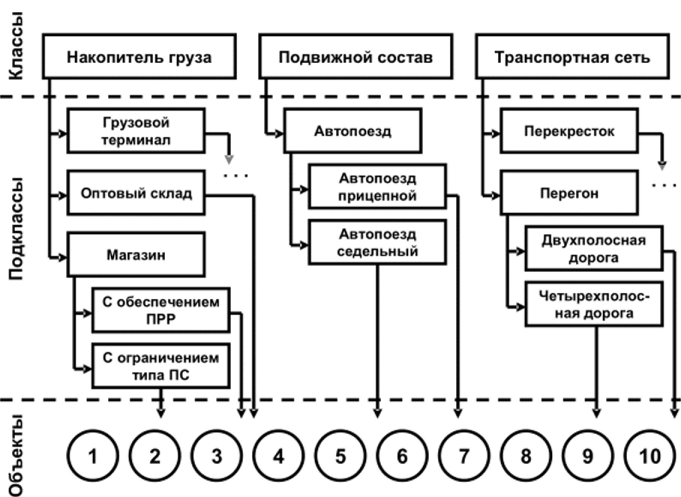


Рисунок 9 – Фрагмент иерархии классов для типичного процесса транспортной системы

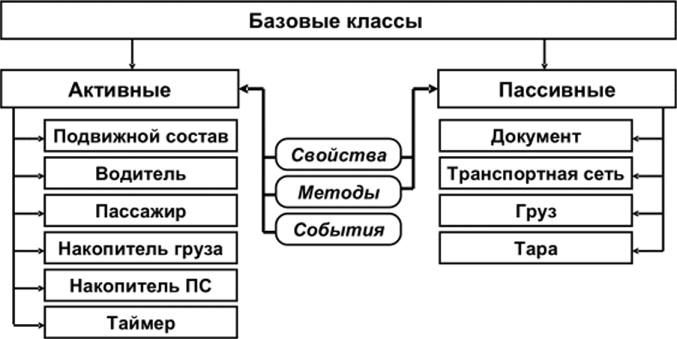


Рисунок 10 – Пример активных и пассивных объектов транспортной системы

Распространение принципов ООМ на планирование и управление транспортной системой позволяет использовать информацию о состоянии объектов управления на основе анализа событий. В зависимости от этого, инициируя выполнение тех или иных методов, можно изменять процесс функционирования системы, добиваясь оптимальных характеристик ее работы. При этом существенные потери времени, связанные с последовательной технологией выполнения процесса в транспортной системе, могут быть сокращены за счет параллельного выполнения методов для различных объектов или группы объектов. [4]

Например, для каждого объекта набор методов можно объединить в четыре группы:

* планирование и (или) подготовка объекта к выполнению каких-либо действий;
* выполнение действия;
* проверка завершенности действия и правильности его выполнения;
* документальное оформление действия.

В этом случае выполнение методов всех четырех групп составит завершенный цикл функционирования объекта, как это показано на рисунке 11.

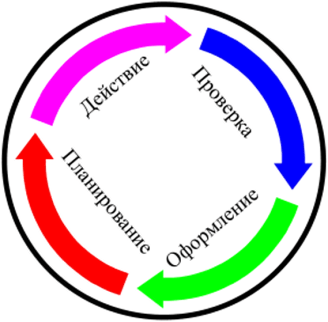


Рисунок 11 – Цикл работы объекта

Параллельность работы системы обеспечивается за счет того, что методы для связанных объектов выполняются не после завершения цикла работы каждого объекта, а могут инициироваться определенными событиями. Схематично это показано на рисунке 12. Реализация предложенного подхода может быть осуществлена с помощью уже сегодня доступных для автоперевозчиков средств телематики.

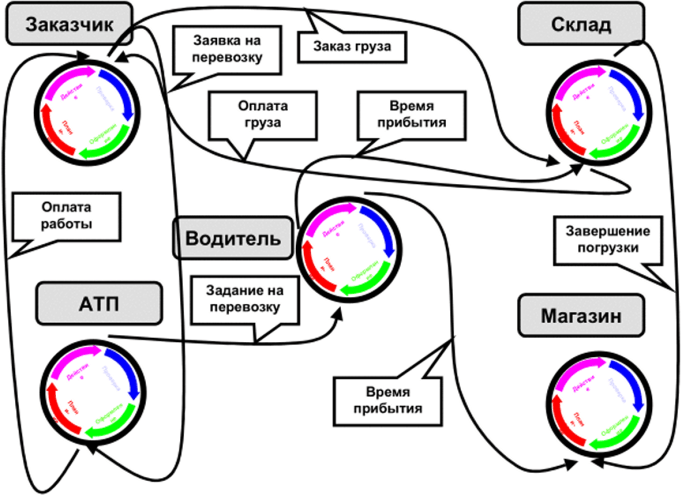


Рисунок 12 – Пример схемы планирования и управления доставкой груза, основанной на принципах ООМ

Таким образом, использование принципов ООМ обеспечивает:

* автоматизацию управления внешними бизнес-операциями между различными субъектами транспортной системы;
* оптимизацию транспортной системы за счет параллельного выполнения отдельных процессов;
* достоверную информацию о состоянии системы в реальном масштабе времени.

# Заключение

Анализ таблицы 2 показывает, что, несмотря на достаточную выразительность языка UML для построения наглядной имитационной модели, его средств явно недостаточно. Приведенные абстрактные объекты имитационной модели составляют концептуальную модель функционирования отдела продаж и маркетинга.

Из примера мы можем увидеть, что использование ООМ помогает провести оптимизацию работы предприятия, осуществляет автоматизацию управления внешними бизнес-операциями, а также предоставляет достоверную информацию о состоянии системы в реальном времени.

# Список используемой литературы

1. Методология объектно-ориентированного моделирования [Электронный ресурс] Режим доступа: <https://studme.org/174095/tehnika/metodologiya_obektno_orientirovannogo_modelirovaniya> (дата обращения: 12.05.2020).

2. Объектно-ориентированное моделирование [Электронный ресурс] Режим доступа: <https://studopedia.org/3-32386.html> (дата обращения: 26.05.2020).

3. Объектно-ориентированный подход при визуальном моделировании [Электронный ресурс] Режим доступа: <https://www.top-technologies.ru/ru/article/view?id=24407> (дата обращения: 26.05.2020).

4. Объектно-ориентированный подход к моделированию транспортных систем [Электронный ресурс] Режим доступа: <https://studme.org/286133/tehnika/obektno_orientirovannyy_podhod_modelirovaniyu_transportnyh_sistem> (дата обращения: 26.05.2020).