

### 3. Диаграммы классов

#### 3.1 Теоретические сведения

Класс представляет собой описание совокупности объектов с общими атрибутами, операциями, отношениями и семантикой.

Графически класс изображается прямоугольником с несколькими полями, как показано на рис. 9. Обычно полей три. Первое поле — это название класса. Второе — атрибуты (члены класса, которые в зависимости от языка программирования также могут называться полями или свойствами). В примере на рисунке класс «Класс1» имеет один атрибут «x» целого типа. Перед атрибутом можно указывать тип его видимости (другое название – модификатор доступа), в данном случае значок «-» означает, что атрибут имеет тип «protected».

Третье поле в изображении класса отведено под операции (члены класса, которые также называются методами, функциями или процедурами). На рисунке это поле пока пустое, но показаны окна, добавляющие операцию в класс.

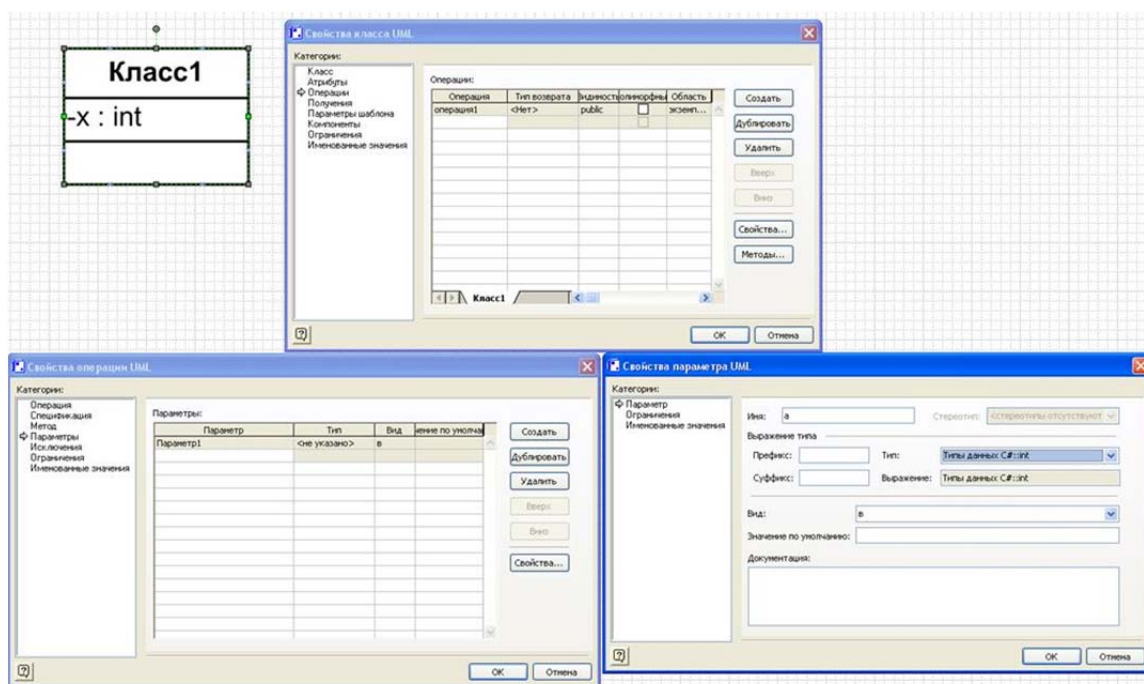


Рис. 9 — Графическое представление класса UML и его свойств в MS Visio

Верхнее окно показывает свойства класса. Как видно в левом поле этого окна, операции (как и атрибуты) являются свойством класса. В правом поле этого окна приведена таблица со списком операций, и кнопки для редактирования списка операций. В этой таблице указаны основные свойства операции, например, тип возврата.

Более подробно свойства операций можно установить, нажав кнопку «свойства» в верхнем окне. Появится окно свойств операции (на рисунке – нижнее левое). В левом поле этого окна приведен список свойств. В примере выбрано свойство параметр операции (другим словами это аргумент функции класса). Основные свойства параметра приведены в таблице по центру, и, аналогично свойствам операции, более подробно свойства параметров можно задать, нажав кнопку свойства (нижнее правое окно).

Подробное описание всех свойств классов заняло бы очень много времени, поэтому рекомендуется их изучать по мере использования, используя в качестве справочной информации книги по объектно-ориентированному программированию.

Совокупность всех классов в информационной системе образует так называемый *словарь* системы. Составление словаря – это наиболее важная и сложная часть проектирования объектно-ориентированной системы. Кроме того, эта часть проекта наиболее часто подвергается переделке. Существует много способов, методов и рекомендации по поводу того, как нужно составлять словарь системы. Мы рассмотрим лишь некоторые приемы по созданию диаграммы классов.

Наиболее известным формальным методом создания словаря системы является метод CRC-карт или таблиц (Class-responsibility-collaboration, класс-ответственность-кооперация или сущность-связь-ответственность). Идея заключается в том, что при проектировании словаря основное внимание должно уделяться сущности класса, т.е. его функциям и отношениям с другими классами, а не деталям реализации, таким как конкретные атрибуты и операции. Поэтому для каждого класса заводится карточка (или строка в таблице), на которой отображаются существенные детали класса: название класса, его ответственность (за что класс отвечает) и кооперация или связь (т.е. с какими классами он связан). В некоторых случаях в карточки заносятся также подклассы, суперклассы и автор этого класса.

Общая схема разработки диаграммы при этом не изменяется. Обычно имеется общее описание системы, например, в виде диаграммы прецедентов. Необходимо составить полный список всех возможных сущностей системы вместе с их назначением. Выделяя общие свойства, все сущности и их ответственности группируются в относительно небольшое число классов. Для классов создаются CRC-карточки или таблицы, в которые заносятся основные свойства классов. Полученная таблица анализируется. Классы в таблице должны покрывать все сущности и ответственности, при этом они должны быть сбалансированы, т.е. иметь примерно одинаковый объем ответственности. В случае несбалансированности нагрузки классов нужно провести перераспределения обязанностей или реформирование классов.

Дополнительные возможности по балансировке нагрузки между сущностями предоставляет механизм наследования в объектно-ориентированном программировании. Если несколько сущностей имеют общие свойства или ответственность, и, в тоже время имеют и различные свойства, не позволяющие их удобно сгруппировать в один класс, то можно выделить суперкласс, включающий все общие свойства, а различия в свойствах сущностей реализовать в классах-наследниках. В этом случае в CRC-таблице имеет смысл завести колонки подклассов и суперклассов.

После получения сбалансированной CRC-таблицы классы можно расположить на диаграмме классов. Отношения между классами берутся из колонки «кооперация» (связь). В этом случае наиболее часто между классами используется отношения «ассоциация». Если есть, то подклассы и суперклассы связываются между собой отношением наследования.

На диаграмме классов для указания характера отношений между классами часто используются дополнения к отношению ассоциация. Примеры дополнения приведены на рис. 10.

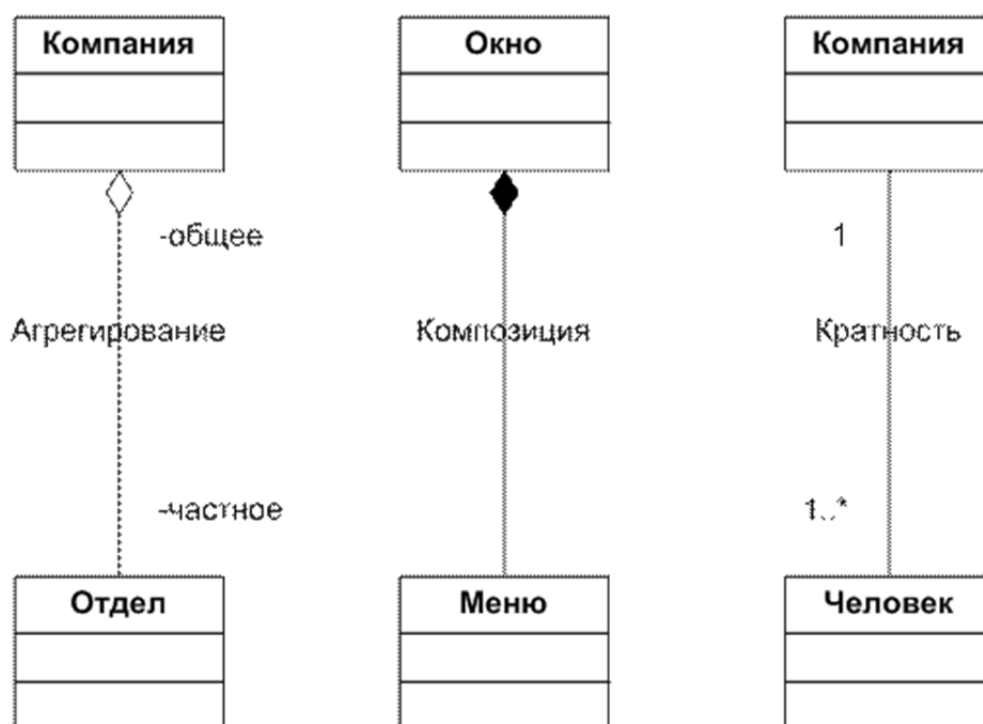


Рис. 10 — Дополнения к отношению ассоциация на диаграмме классов

Агрегирование показывает отношения «часть/целое» между двумя классами, например, отдел является частью компании. Композиция показывает, что один класс входит в другой, например, в окна графического интерфейса обычно входит меню. В MS Visio это дополнение выставляется в свойствах отношения, в колонке агрегат. При моделировании баз данных указывается кратность ассоциации.

После составления словаря для каждого класса определяются атрибуты и операции. На этом этапе проектирования информационной системы не нужно детально выписывать всех членов классов. Более точное описание класса удобно производить на последующих диаграммах, на которых рассматривается работа объектов этого класса.

### 3.2 Методика построения диаграммы классов

1. Открыть созданный ранее файл UML.
2. В проводнике по модели UML щелкнуть правой кнопкой мыши по папке «Основной пакет», выбрать команду меню «Создать» и далее «Структурная схема».
3. У рабочего листа MS Visio появится название «Структурная схема-1». Как говорилось ранее, вся модель UML с множеством схем будет сохранена в одном файле, поэтому необходимо давать соответствующим листам более короткие названия. Для переименования созданного листа щелкнуть правой кнопкой мыши по его ярлычку, выбрать команду «Переименовать» и задать имя, например, ДКл (сокращенно от диаграмма классов).
4. В результате создания новой диаграммы классов, автоматически откроется соответствующий шаблон графических элементов для данной диаграммы.
5. Открыть команду меню UML и нажать на кнопку «Стереотипы». В появившемся окне создать три новых стереотипа «Сущность», «Граница» и «Управление». Для всех трех вновь созданных стереотипов задать в качестве базового класса «Класс».
6. Разместить на рабочем листе в необходимом количестве элементы «Класс» и дать им соответствующие названия. Для этого нужно дважды щелкнуть по классу и в появившемся окне свойств в категории «Класс» в поле «Имя» ввести название класса.
7. Задать для классов атрибуты и операции. Для этого также нужно вызвать окно свойств класса и выбрать в нем справа соответствующие категории «Атрибуты» и «Операции». Для атрибутов желательно задать типы данных. Параметры «Видимость», «Кратность» и «Начальное значение» можно не задавать. Для операций также можно не задавать параметры. Для отключения отображения необязательных параметров класса, нужно щелкнуть по нему правой кнопкой мыши и выбрать команду меню «Параметры отображения фигуры ...». В появившемся окне установить нужные параметры.

### 3.3 Пример построения диаграммы классов

Построим диаграмму классов для системы продажи товаров по каталогу. Сначала составим словарь системы. Для этого построим таблицы сущность – связь - ответственность:

|                             |                    |
|-----------------------------|--------------------|
| <b>Имя класса:</b> Продавец | <b>Суперкласс:</b> |
| <b>Подкласс:</b>            |                    |
| <b>Ответственность:</b>     |                    |
| <b>Кооперация:</b> Товар    |                    |

|                                    |                    |
|------------------------------------|--------------------|
| <b>Имя класса:</b> Товар           | <b>Суперкласс:</b> |
| <b>Подкласс:</b>                   |                    |
| <b>Ответственность:</b>            |                    |
| <b>Кооперация:</b> Продавец, Заказ |                    |

|                                       |                    |
|---------------------------------------|--------------------|
| <b>Имя класса:</b> Заказ              | <b>Суперкласс:</b> |
| <b>Подкласс:</b>                      |                    |
| <b>Ответственность:</b>               |                    |
| <b>Кооперация:</b> Товар, ЗаказОплата |                    |

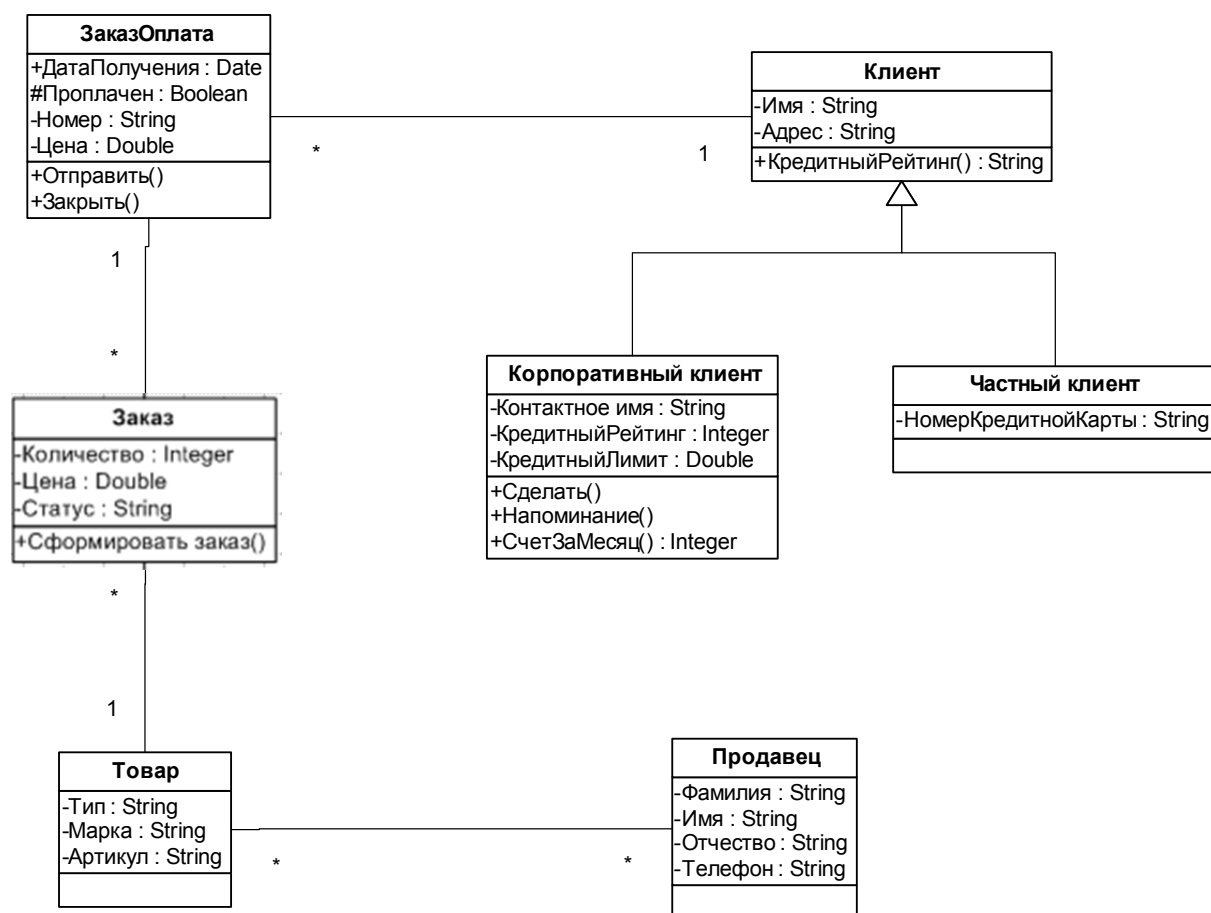
|   |                    |
|---|--------------------|
| <b>Имя класса:</b> ЗаказОплата                          | <b>Суперкласс:</b> |
| <b>Подкласс:</b>  |                    |
| <b>Ответственность:</b> Отправить оплату, Закрыть заказ |                    |
| <b>Кооперация:</b> Заказ, Клиент                        |                    |

|   |                    |
|---|--------------------|
| <b>Имя класса:</b> Клиент                             | <b>Суперкласс:</b> |
| <b>Подкласс:</b> Корпоративный клиент, Частный клиент |                    |
| <b>Ответственность:</b> наличие Кредитного рейтинга   |                    |
| <b>Кооперация:</b> ЗаказОплата                        |                    |

|  |                           |
|--|---------------------------|
| <b>Имя класса:</b> Корпоративный клиент  | <b>Суперкласс:</b> Клиент |
| <b>Подкласс:</b>   |                           |
| <b>Ответственность:</b> Сделать заказ, Напоминание о статусе заказа, проверить счет за месяц |                           |
| <b>Кооперация:</b>   |                           |

|                                   |                           |
|-----------------------------------|---------------------------|
| <b>Имя класса:</b> Частный клиент | <b>Суперкласс:</b> Клиент |
| <b>Подкласс:</b>                  |                           |
| <b>Ответственность:</b>           |                           |
| <b>Кооперация:</b>                |                           |

Таким образом, получаем классы: Продавец, Товар, Заказ, ЗаказОплата, Клиент. Итоговая диаграмма классов представлена на рисунке 11.



**Рисунок 11 — Диаграмма классов для системы продажи товаров по каталогу**

Диаграмма классов в дальнейшем может уточняться.

## 4. Диаграммы последовательностей

### 4.1 Теоретические сведения

В UML существует 5 видов диаграмм, которые отображают динамические свойства системы: последовательностей и коопераций (эти два типа называются диаграммами взаимодействий), деятельности, состояний и прецедентов. В нашем курсе рассмотрим диаграммы *последовательностей*.

Эти диаграммы описывают взаимодействия множества объектов, включая сообщения, которыми они обмениваются.

В отличие от диаграммы классов, на которой изображаются абстрактные элементы в виде классов, на диаграмме последовательностей используются конкретные экземпляры классов – *объекты*. Объекты отображаются прямоугольником без полей. Для того чтобы подчеркнуть, что это экземпляр абстрактной сущности, название объекта подчеркивается. При необходимости через двоеточие после названия можно указать сущность (класс) экземпляром которой является этот объект. Отметим, что объект может быть экземпляром не только класса, но и других абстракций, например, актера. Обратите внимание, что при указании в качестве классификатора актера изменится графическое обозначение объекта (рис. 12).



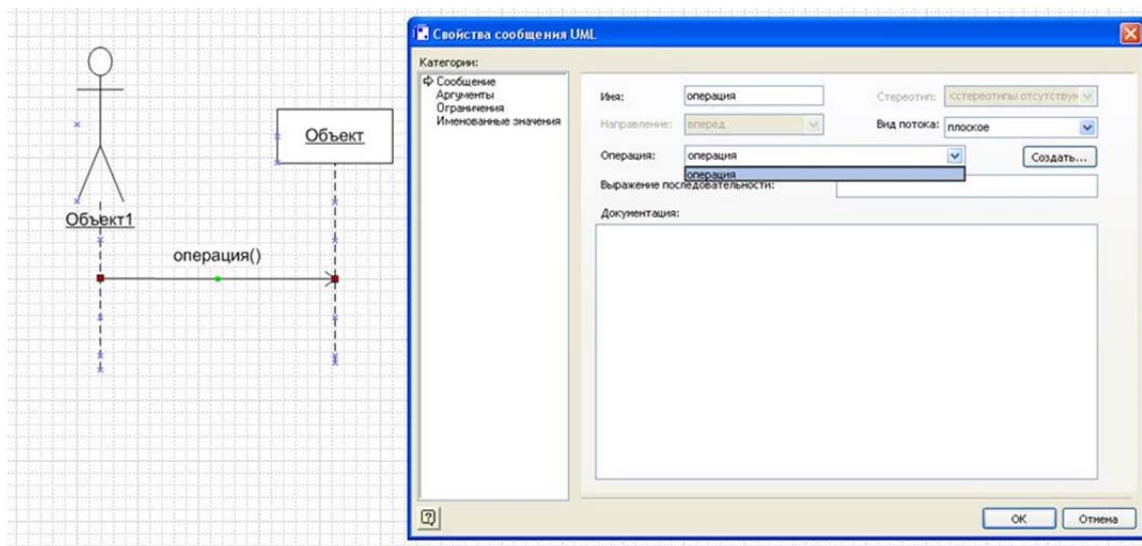
Рисунок 12 — Графическое обозначение объекта UML

На диаграмме последовательностей у объекта может присутствовать *линия жизни*, на которой отмечаются происходящие с объектом события. Линия жизни отображается пунктирной линией (рис. 13).

Между собой объекты могут быть связаны связями. *Связь* – это экземпляр отношения ассоциация, и имеет такое же графическое обозначение, что и ассоциация. Они применяются, обычно, на диаграмме коопераций.

На диаграмме последовательностей объекты обмениваются сообщениями. *Сообщение* – это спецификация передачи данных от одного объекта другому, который предполагает какое-то ответное действие. Графически сообщение обозначается сплошной линией со стрелкой.

Часто операция вызывает какую-либо операцию в объекте. Очевидно, что класс, экземпляром которого является объект, должен иметь такую операцию. Привязка сообщения к операции класса объекта выполняется в свойствах сообщения (рис. 13).



**Рисунок 13 — Сообщение UML и его свойства**

При построении диаграммы классов обычно определяются только основные свойства сущностей, а такие детали, как операции, удобно создавать при построении диаграммы последовательности, для чего в свойствах сообщения UML есть кнопка создания операции.

Диаграммы последовательностей, как и другие диаграммы для отображения динамических свойств системы, могут быть выполнены в контексте многих сущностей UML. Они могут описывать поведение системы в целом, подсистемы, класса или операции класса и др. К сожалению, Visio не достаточно гибка в плане поддержки раскрытия содержания отдельных элементов с помощью других диаграмм. Например, кликнув правой кнопкой мыши по классу можно обнаружить, что для его описания можно создать лишь диаграммы классов, состояний и деятельности. Поэтому возможность привязать диаграмму последовательностей к элементу, который она реализует, средствами Visio не возможно, эту связь нужно подразумевать.

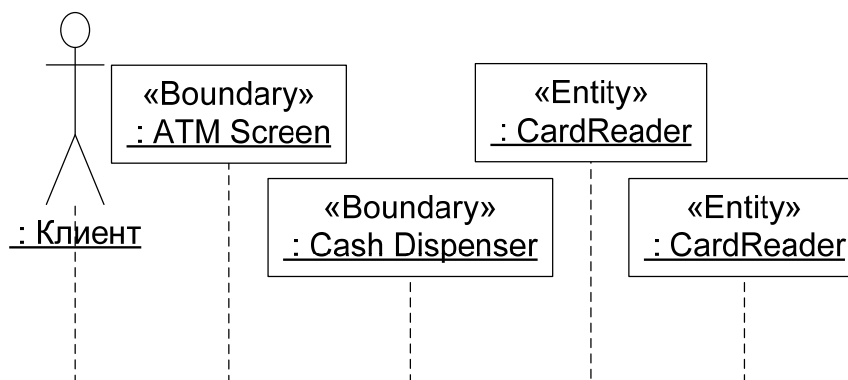
Диаграммы последовательностей будем делать в контексте прецедентов с диаграммы прецедентов, реализуя те функции, которые должна выполнять наша система.

При построении динамических диаграмм используется уже разработанная структура информационной системы. Для диаграммы последовательностей не нужно придумывать объекты, а достаточно определить, экземпляры каких классов участвуют в этом действии.

Определив необходимые объекты (как экземпляры классов, так и экземпляры актеров), вторым этапом построения диаграмм последовательностей определяются сообщения, пересылаемые между актерами. Фактически определяется последовательность шагов, для выполнения нужного действия.

#### **4.2 Методика построения диаграммы последовательностей**

1. Открыть созданный ранее файл UML.
2. В проводнике по модели UML щелкнуть правой кнопкой мыши по папке «Основной пакет», выбрать команду меню «Создать» и далее «Схема последовательностей».
3. У рабочего листа MS Visio появится название «Последовательность-1». Переименовать созданный лист, дав ему имя ДП (сокращенно от «Диаграмма последовательности»).
4. Разместить в верхней части рабочего листа элемент «Линия жизни». Далее щелкнуть по нему дважды, вызвав окно свойств и в поле «Классификатор» выбрать соответствующего актера, инициирующего рассматриваемый ВИ.
5. Разместить в верхней части рабочего листа в необходимом количестве остальные классы, участвующие в реализации данного ВИ и соотнести их с соответствующими классами. Если на диаграмме размещено много объектов и они не умещаются по ширине можно использовать многоуровневое расположение объектов, как показано на рис. 14.



**Рисунок 14. Многоуровневое расположение объектов на диаграмме последовательности**

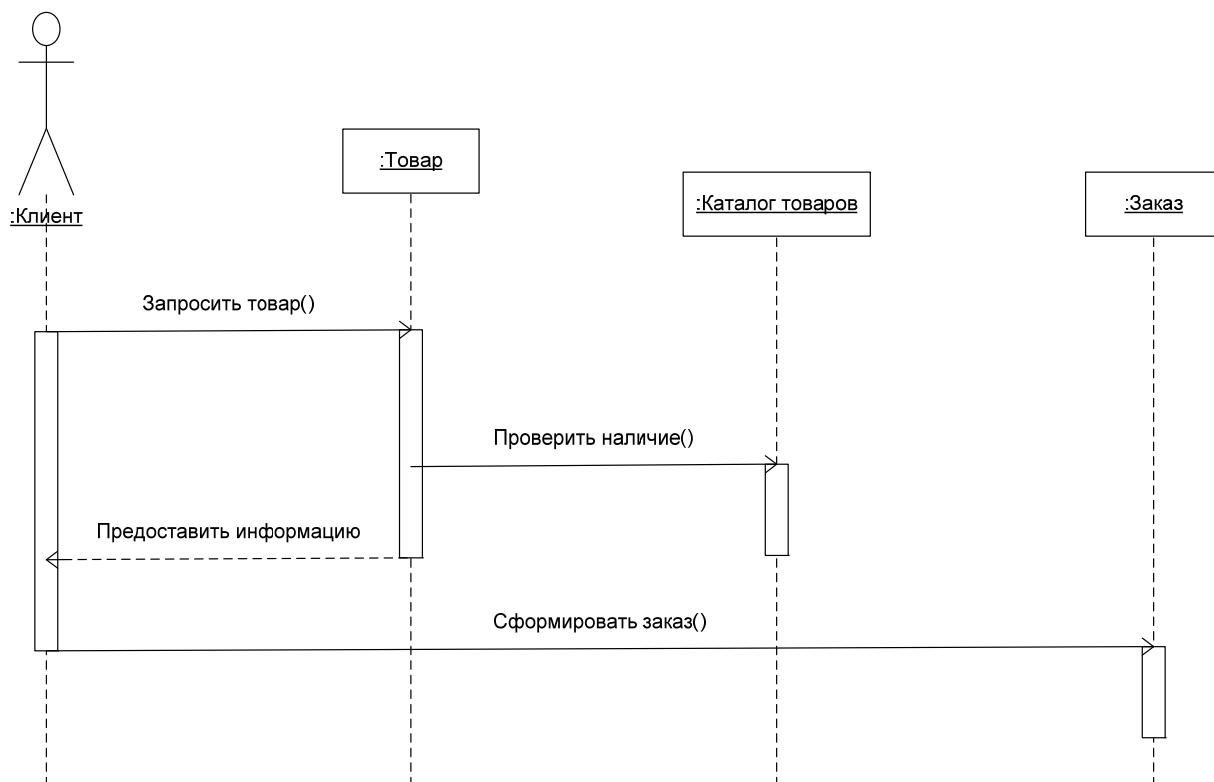
6. Растянуть вниз пунктир линии жизни. Соединить классы между собой с помощью элемента «Сообщение». При этом линия не всегда будет параллельна горизонтальной оси, так как соединяются фигуры через точки соединения, а они не всегда расположены симметрично по отношению друг к другу. В этом нет ничего страшного, главное чтобы окончание линии находилось по уровню не выше ее начала.
7. Сопоставить сообщение, которое получает объект для исполнения с одной из его операций, заданной в предыдущей лабораторной работе. Для этого нужно вызвать окно свойств для ассоциации (дважды щелкнув по нему). В появившемся окне в поле «Операция» выбрать из списка соответствующую операцию. Для того, чтобы операция присутствовала в списке, она должна была быть задана для данного класса в лабораторной работе 3. Можно добавить для класса необходимую операцию прямо из этого окна, если нажать на кнопку «Создать».

#### **4.3 Пример построения диаграмм последовательностей**

Построим диаграмму последовательности для варианта использования «Обеспечить покупателя информацией» (рис. 15). Для этого добавим на диаграмму последовательности линии жизни и соотнесем объекты с актерами, инициирующими вариант использования «Обеспечить покупателя информацией», и с необходимыми классами.

Добавим сообщения, которыми обмениваются объекты для исполнения варианта использования.

Если объект имеет операцию (посмотреть в лабораторной работе «Диаграммы классов» наличие операции у класса, которому принадлежит объект).

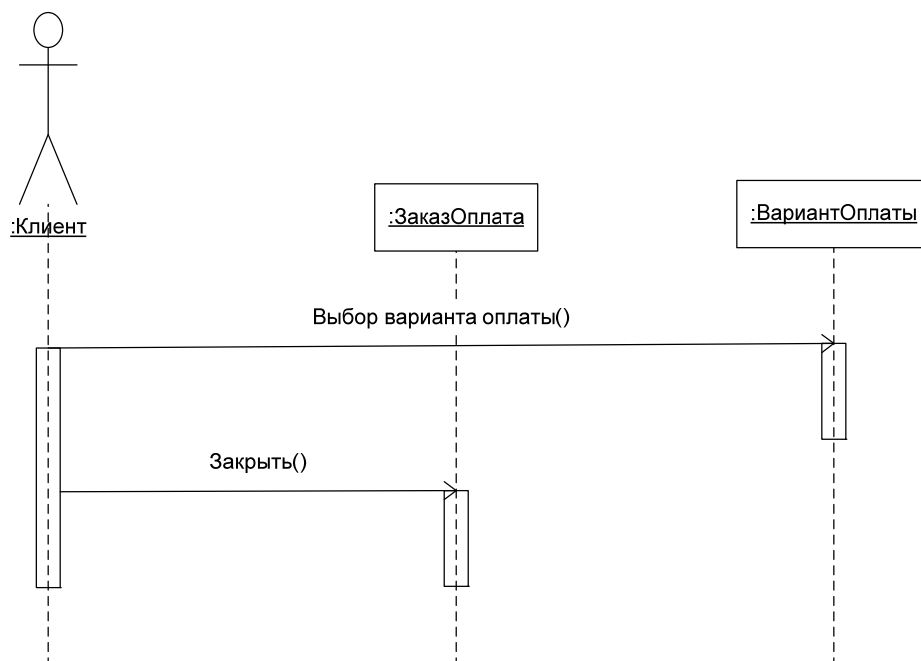


**Рисунок 15 — Диаграмма последовательности  
для варианта использования «Обеспечить покупателя информацией»**

При построении диаграмм последовательностей можно вносить коррективы в диаграмму классов. Если объект класса получает новую операцию, то она добавляется в соответствующий класс на диаграмме классов как метод.

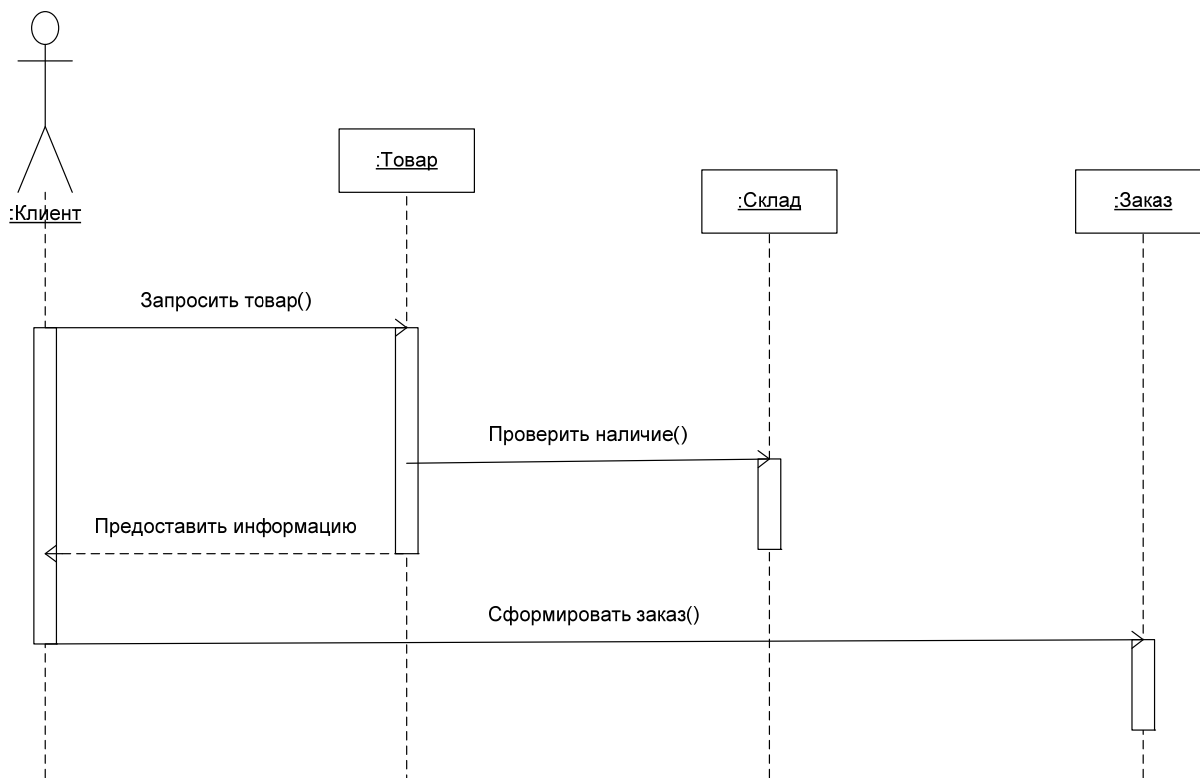
Построим диаграмму последовательности для варианта использования «Согласовать условия оплаты» (рис. 16). Действия по построению диаграммы аналогичны построению диаграммы последовательности для варианта использования «Обеспечить покупателя информацией». Если при создании объекта необходимый классификатор отсутствует, то его необходимо создать в окне свойств роли классификатора UML.





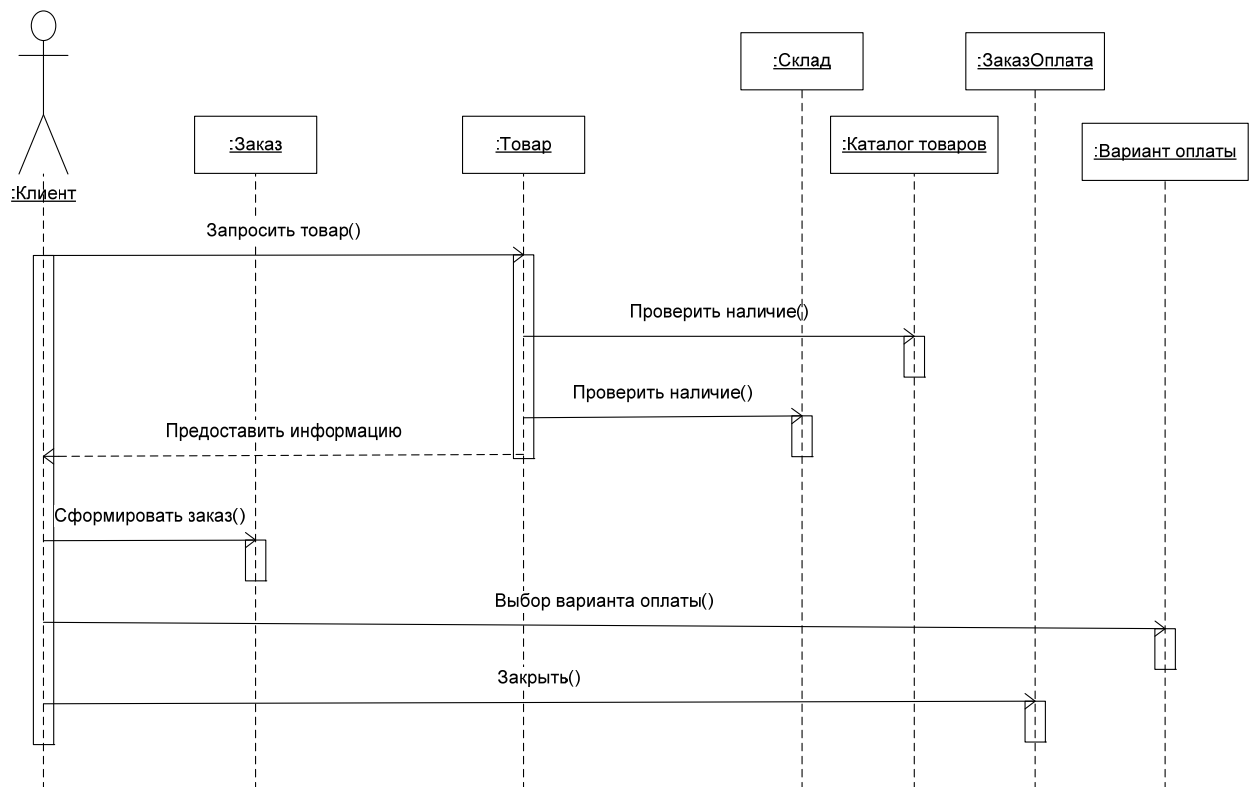
**Рисунок 16 — Диаграмма последовательности для варианта использования «Согласовать условия оплаты»**

Построим диаграмму последовательности для варианта использования «Заказать товар со склада» (рис. 17). Действия по построению диаграммы аналогичны построению предыдущих диаграмм последовательностей.



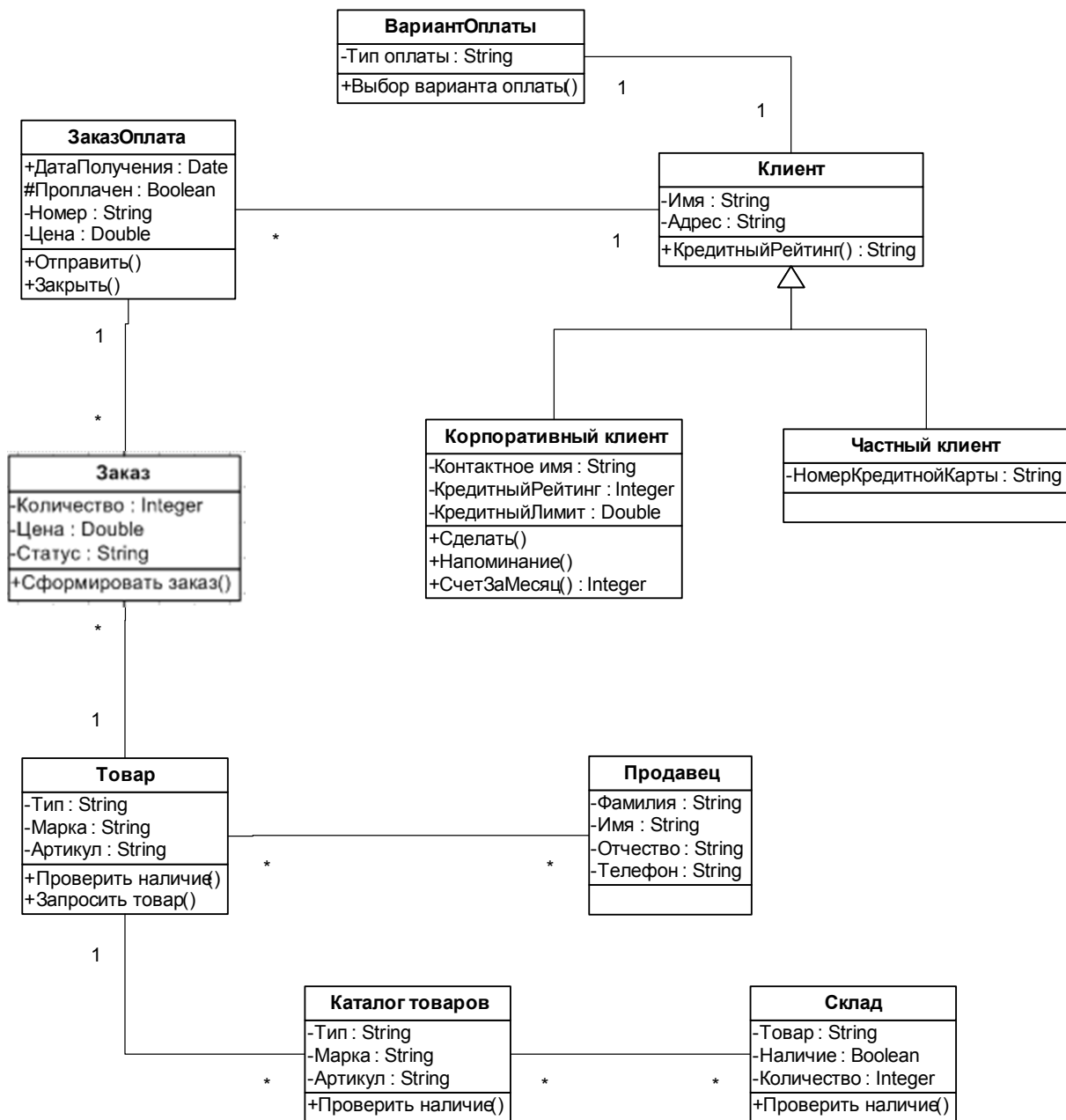
**Рисунок 17 — Диаграмма последовательности для варианта использования «Заказать товар со склада»**

Построим диаграмму последовательности для системы продажи товаров по каталогу (рис. 18).



**Рисунок 18 — Диаграмма последовательности для системы продажи товаров по каталогу**

Изменим диаграмму классов в соответствии с созданными диаграммами последовательностей (рис. 19).



**Рисунок 19 — Уточненная диаграмма классов для системы продажи товаров по каталогу**