

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Рязанский государственный радиотехнический университет  
имени В. Ф. Уткина»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ  
ПО ТЕХНОЛОГИЯМ ПРОЕКТИРОВАНИЯ И РАЗРАБОТКЕ  
СПЕЦИАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ТРЕБОВАНИЯ К  
ОФОРМЛЕНИЮ ОТЧЁТОВ**

Версия: Raw incomplete uc – act – stm

**Рязань 2024**

## Оглавление

Общие замечания.....	3
1 Лабораторная работа «ДИАГРАММА вариантов использования» .....	3
1.1 Ассоциированные понятия .....	3
1.2 Руководящие принципы построения диаграммы вариантов использования .....	3
1.3 Требования к оформлению отчёта .....	5
2 Лабораторная работа «Диаграмма деятельности» .....	5
2.1 Ассоциированные понятия .....	6
2.2 Руководящие принципы построения диаграммы деятельности .....	6
2.2.1 Оформление ветвлений.....	6
2.2.2 Оформление разделов активности (Activity Partitions) в нотации плавательных дорожек (Swimlanes).....	7
2.3 Требования к оформлению отчёта .....	9
3 изучаемая тема «Диаграмма автомата» .....	10
3.1 Ассоциированные понятия .....	10
3.2 Руководящие принципы построения диаграммы автомата.....	10
3.2.1 Общие замечания.....	10
3.2.2 Принцип иерархии / декомпозиции в диаграммах автомата .....	11
3.2.3 Примеры диаграмм автомата .....	11
Литература.....	15

## ОБЩИЕ ЗАМЕЧАНИЯ

Все создаваемые диаграммы должны быть тщательно документированы. Документированию подлежит сама диаграмма (секция Documentation заполняется при отсутствии выделений элементов диаграммы) и все её элементы – документируются все узлы (геометрические фигуры) диаграммы и все стрелки.

Материалы документирования представляются в отчетах по лабораторным работам. Эти материалы желательно подразделять на структурные части с такими названиями как «Документирование акторов», «Документирование прецедентов», «Документирование действий и деятельности», «Документирование состояний», «Документирование отношений», «Документирование переходов».

Обратим внимание на то, что теоретические сведения и методические указания в отчётах приводить не следует. Теорию необходимо знать, а требования, содержащиеся в методических указаниях, необходимо выполнять.

### 1 ИЗУЧАЕМАЯ ТЕМА «ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ»

Практическое задание по данной теме выполняется с использованием [2]. Изучаются материалы четвертой и пятой глав.

#### 1.1 Ассоциированные понятия

Вариант использования = Use Case = Прецедент – это некоторая  
Функция = Задача = Работа = Процесс, реализующий Функцию (Задачу, Работу)  
На диаграмме прецедент – это овал с названием Функции (Задачи, Работы).

Осознаём и запоминаем:  
формулировка Функции (Задачи, Работы) одновременно является и  
**названием** Процесса, реализующего Функцию (Задачу, Работу)

Например, «Решение алгебраического уравнения» – это одновременно и формулировка задачи, и название процесса решения уравнения.

По этой причине диаграммы вариантов использования можно называть и функциональными, и процессными (поведенческими). Дополнительным основанием называть диаграммы вариантов использования поведенческими является то, что для прецедентов создаются потоки (*текстовые алгоритмы*) реализации, размещаемые в текстовых файлах, прикрепляемых к прецедентам редактором вложений в окне закладки диаграммы с именем Attachment.

#### 1.2 Руководящие принципы построения диаграммы вариантов использования

Диаграмма вариантов использования – это, по сути дела, не одна, а *несколько* диаграмм: одна из них – основная, а остальные – декомпозиционные, – структурирующие варианты использования, представленные на основной диаграмме.

Основная диаграмма содержит относительно независимые функции моделируемой системы – варианты её использования. Таких вариантов в учебных проектах, как правило, бывает немного – от трёх до девяти. Например, «Регистрация клиента», «Вход клиента в систему», «Поиск и выбор товара», «Оформление продажи», «Учет клиентов, товаров и продаж». На основной диаграмме не принято осуществлять даже частичную декомпозицию вариантов использования. При правильном определе-

нии вариантов использования ИС на основной диаграмме, как правило, отсутствуют отношения зависимости между вариантами, что свидетельствует о независимости этих вариантов. В приведённом выше примере все варианты использования *функционально* (алгоритмически) независимы. Они связаны только *информационно* – «Учет клиентов, товаров и продаж» использует *данные*, порождённые «Регистрацией клиента» и «Оформление продажи». Данные на диаграммах вариантов использования не отображаются основными изобразительными средствами, но крайне полезно показать информационную взаимосвязь вариантов с помощью средств пояснения – аннотациями типа «Text», «Note» (предпочтительно) и «NoteLink» (для указания к чему относится Note – замечание). Пример информационного дополнения диаграммы вариантов средствами аннотирования приведён на рисунке 1.



Рисунок 1 – Пример информационного дополнения диаграммы вариантов использования средствами аннотирования

Декомпозиционные диаграммы *должны* содержать декомпозируемый прецедент, – образующий логический центр декомпозиционной диаграммы, и связанный с прочими её прецедентами отношением зависимости. Пример декомпозиции варианта использования показан на рисунке 2.

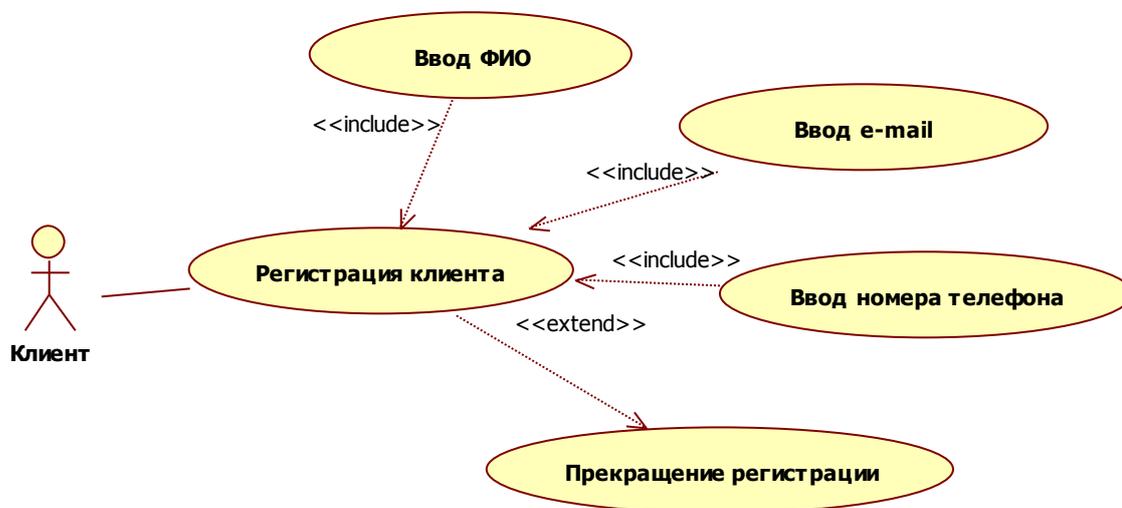


Рисунок 2 – Пример декомпозиции варианта использования

В методичке Каюмовой на рисунке 21 это правило не выполнено – основной прецедент «Заказ товаров» на диаграмме отсутствует, в результате чего вспомогательные прецеденты превращены в основные, поскольку оказались ассоциированными с актором «Покупатель».

### 1.3 Требования к оформлению отчёта

1. Все элементы диаграмм – и узлы, и стрелки – должны иметь пояснения в секции Documentation.
2. Пояснение стрелки должно представлять собой *обоснования* выбранного отношения, представляемого данной стрелкой.
3. Текст пояснений в отчёте желательно подразделять на структурные части – «Документирование акторов», «Документирование прецедентов», «Документирование отношений (стрелок)»

#### Примерные схемы обоснований:

*Для ассоциации между актором и прецедентом:*

а) «Отношение ассоциации выбрано направленным от актора <название актора> к прецеденту <название прецедента>, потому что актор является пассивным участником взаимодействия – запускает выполнение прецедента, задаёт исходные данные и получает результат».

б) «Отношение ассоциации между актором <название актора> и прецедентом <название прецедента> выбрано ненаправленным, потому что актор является активным участником взаимодействия – не только запускает выполнение прецедента, задаёт исходные данные и получает результат, но и участвует в диалоге с программой, реализующей прецедент».

*Для зависимости между прецедентами:*

а) «Тип отношение зависимости между прецедентами <название 1-го прецедента> и <название 2-го прецедента> выбран «include», потому что при реализации прецедента <название 1-го прецедента> функциональность прецедента <название 2-го прецедента> используется безусловно.

б) «Тип отношение зависимости между прецедентами <название 1-го прецедента> и <название 2-го прецедента> выбран «extend», потому что при реализации прецедента <название 1-го прецедента> функциональность прецедента <название 2-го прецедента> используется только при выполнении условия <формулировка условия>».

*Замечание:* условие <формулировка условия> обязательно помещается в свойство (property) Condition поясняемой стрелки отношения «extend».

4. Все варианты использования, представленные на диаграмме первого уровня (основной диаграмме), должны иметь текстовый алгоритм реализации (спецификацию потоков) в текстовом документе, прикрепляемом редактором вложений в окне закладки с именем Attachment.

## 2 ИЗУЧАЕМАЯ ТЕМА «ДИАГРАММА ДЕЯТЕЛЬНОСТИ»

Практическое задание по данной теме выполняется с использованием [2]. Изучается материалы шестой главы.

## 2.1 Ассоциированные понятия

Поведение ИС – совокупность процессов, реализуемых в ИС.

Процесс (дискретный) – последовательность операций.

Целенаправленные процессы – алгоритмы и сценарии.

1. State machine Diagram = State chart Diagram = Диаграмма автомата

2. Activity Diagram = Диаграмма деятельности.

3. Sequence Diagram = Диаграмма последовательности

4. Communication Diagram = Collaboration Diagram = Диаграмма коммуникации / кооперации (сотрудничества)

D3 = D4 – диаграммы последовательности и коммуникации семантически эквивалентны.

D1  $\supseteq$  D2 – диаграммы автомата и деятельности имеют общую теоретическую платформу – Сети Петри; по совокупности изобразительных средств диаграммы деятельности являются подмножеством диаграмм автомата.

Действие = Action – краткосрочная операция, *непрерываемая* операция

Деятельность = Activity – последовательность действий, длительная, *прерываемая*.

Деятельность может быть конечной или бесконечной во времени.

Бесконечная деятельность = ожидание, возможно, с отображением инструкций или рекламы.

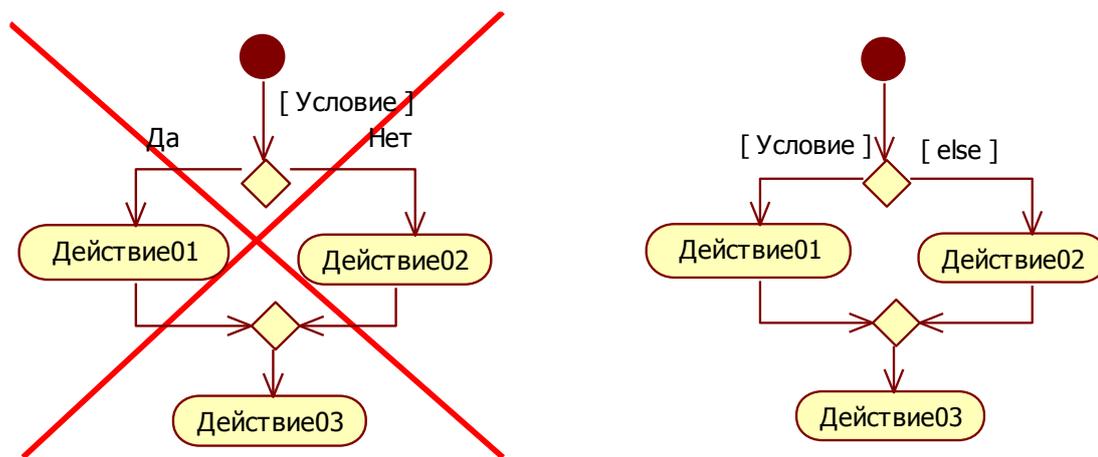
## 2.2 Руководящие принципы построения диаграммы деятельности

Диаграммы деятельности создаются для *каждого* варианта использования, представленного на основной диаграмме прецедентов. Как и диаграмма вариантов использования, диаграмма деятельности – это не одна, а *несколько* диаграмм: одна из них – основная, а остальные – декомпозиционные, – структурирующие *деятельности*, представленные на основной диаграмме и других декомпозиционных диаграммах.

### 2.2.1 Оформление ветвлений

При построении диаграмм деятельности следует избегать ошибки, содержащейся в некоторых методических указаниях, например, в [1] и в интернет-источниках – использования слов «Да» и «Нет» над стрелками, исходящими из узла решения. Эта практика является вольным переносом нотации блок-схем алгоритмов в UML, что совершенно недопустимо. Дело в том, что в UML узел решения – это будущий оператор switch в С-подобных или case в Паскале-подобных языках. Над исходящими стрелками следует размещать условия, при этом над одной из стрелок можно разместить английское else. Следует знать, что условия и else набираются в свойствах (Properties) стрелок, а именно в свойствах «GuardCondition», называемых в некоторых реализациях UML просто «Condition» или «Guard». Набираемые в свойствах условия появляются над стрелками автоматически и в квадратных скобках. Изложенное замечание наглядно иллюстрируется рисунком 3.

Рисунок 3 иллюстрирует также и тот факт, что узел решения в UML является не только средством ветвления, но и средством слияния стрелок. В блок-схемах такого средства нет – стрелки соединяются простым соприкосновением.



а) не верное

б) верное

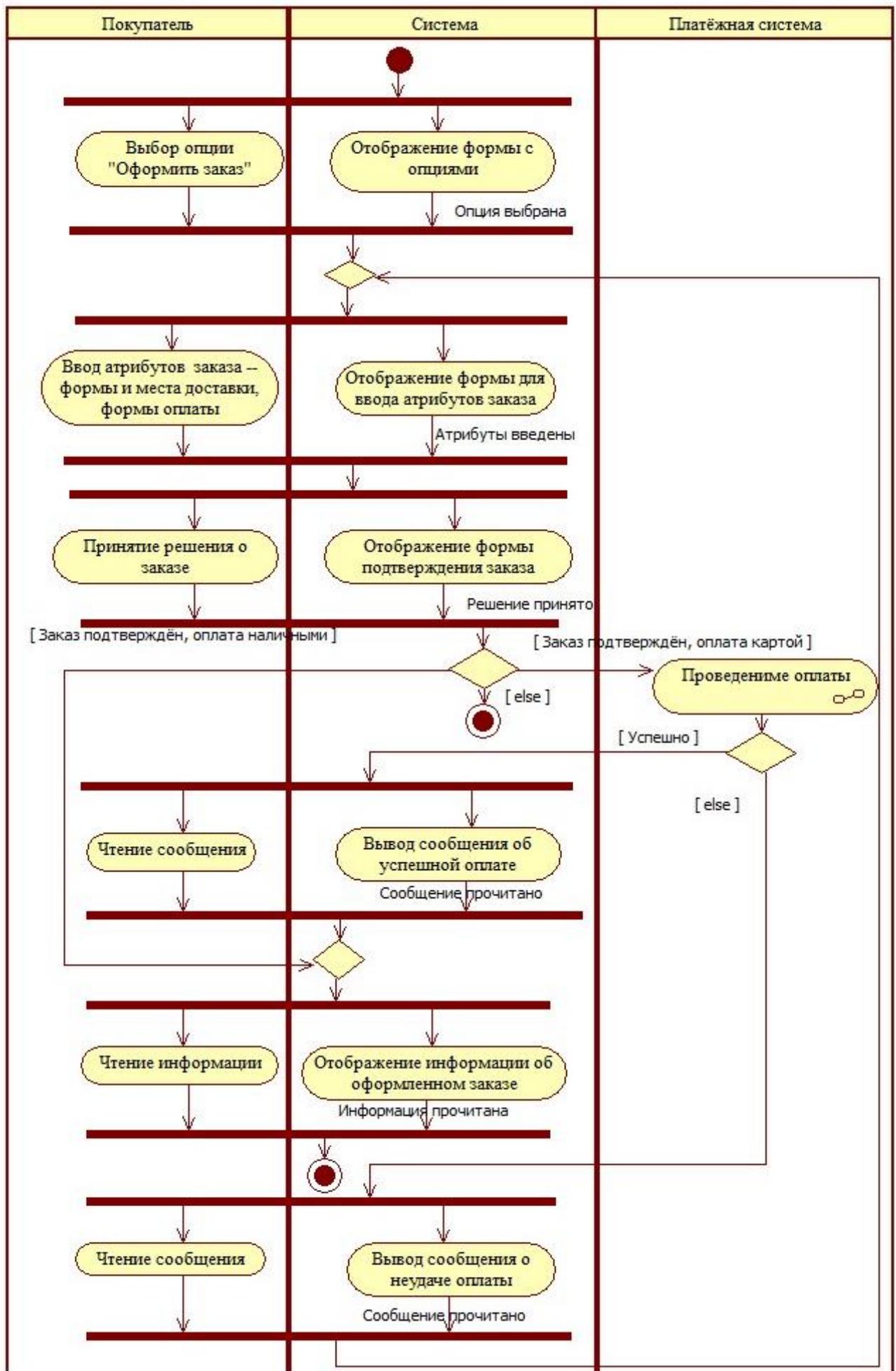
Рисунок 3 – Изображение ветвления в UML

### 2.2.2 Оформление разделов активности (Activity Partitions) в нотации плавательных дорожек (Swimlanes)

При использовании в диаграмме разделов активности в виде плавательных дорожек следует иметь в виду следующее.

Для исполнителей деятельности, являющихся акторами-людьми, следует учитывать, что человек, работая с любой программой может по сути дела (физически) только нажимать на клавиши, пользоваться мышкой, думать и принимать решения. В то же время, в диаграммах деятельности принято указывать не «физику» манипуляций человека, а логическое содержание действий, выполняемых этим человеком – «Ввод исходных данных», «Ввод критерия поиска», «Переход к разделу каталога», «Исследование товара», «Помещение товара в корзину» и т. п. Однако, при этом, крайне желательно представлять в диаграмме механизм, используемый человеком для реализации необходимых действий, например, интерфейсное окно, отображаемое программной системой, как это показано на рисунке 4, представляющем собой пример диаграммы деятельности с разделами активности в виде плавательных дорожек.

Следует также иметь в виду, что «Система», показанная на рисунке 4, интегрирует в себе и серверную и клиентскую (в виде браузера) части программно-информационной системы. Это позволяет избежать излишней детализации действий клиенткой части приложения, не приводя такие действия как «Получение данных», «Рендеринг окна» и т. п. Такой подход вполне приемлем, если он соответствует текущему контексту моделирования.



### 2.3 Требования к оформлению отчёта

1. Все элементы диаграмм – и узлы, и стрелки – должны иметь пояснения в секции Documentation.
2. Пояснение узлов должно устранять неполноту (incompleteness [ɪnkəm'pli:tɪnis]) и неоднозначность (ambiguity [æmbrɪ'ɡju:ɪti]) содержания действий и деятельностей, представленного в названии действия / деятельности. Как правило, одно название не может полностью и однозначно выразить смысл действия / деятельности. Следует стремиться к тому, чтобы пояснения действий и деятельностей, помещаемые в Documentation, обеспечивали представление поясняемых действий и деятельностей в виде программного кода – это требование модели поведения на этапе окончательного (рабочего) проектирования. Модель требований может содержать более размытые (fuzzy ['fʌzi]) формулировки действий и деятельностей, но, тем не менее, они должны обеспечивать точное представление «физики» реализации действия / деятельности.
3. Каждая ссылка (SubactivityState) на деятельность (Activity), для полного представления смысла которой недостаточно текстового пояснения, должна иметь собственную диаграмму деятельности с алгоритмом реализации деятельности, на которую осуществлена ссылка.
4. Пояснение стрелки должно представлять собой обоснования выбранного набора надписей над данной стрелкой – свойств Triggers, Guard Condition, Effects.

#### Примерные схемы обоснований:

*Для ссылок на деятельности:*

а) деятельность «Ожидание ввода данных» (см. рисунок 5) семантически тривиальна, поэтому отдельная диаграмма её реализации не приводится;

б) деятельность «Ввод исходных данных» (см. рисунок 5) предполагает реализацию стандартной процедуры «Ввод с клавиатуры» | реализацию стандартной процедуры «Ввод из файла» | реализацию стандартного процесса заполнения текстовых полей окна пользовательского интерфейса и выбора элементов раскрывающихся списков. [Среди элементов, перечисленных через знак «|», следует выбрать что-то одно – это и будет конкретным проектным решением.] Данная деятельность семантически проста, поэтому отдельная диаграмма её реализации не приводится.

Заметим, что применение к деятельности терминов «тривиальна», «проста» должно быть обоснованным – подобным приведённым выше примерам.

*Для стрелки перехода:*

а) стрелка перехода от действия «Вывод приглашения к вводу ИД» к деятельности «Ожидание ввода данных» не содержит никаких надписей, потому что переход осуществляется по завершению деятельности «Вывод приглашения к вводу ИД» и является безальтернативным, т. е. единственным (см. рисунок 5).

б) из деятельности «Ожидание ввода данных» имеется два выхода, соответствующих двум триггерным событиям – «Нажато CTRL+C» и «Нажата информационная клавиша»; соответствующие переходы осуществляются безусловно и без действий на переходах (см. рисунок 5)

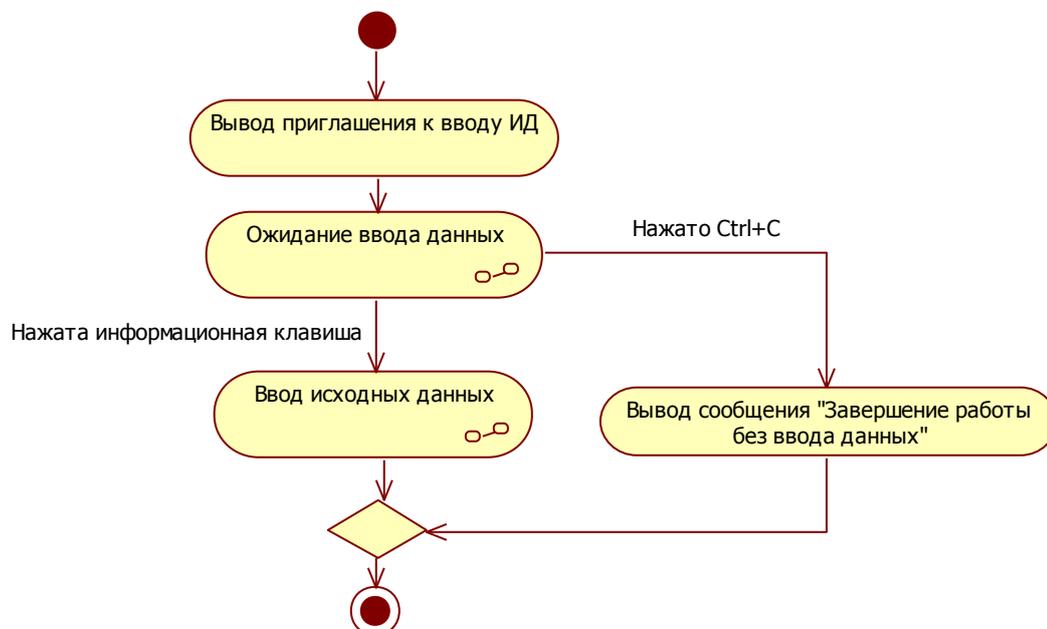


Рисунок 5 – Пример выхода из состояния с бесконечной деятельностью по триггерным событиям

### 3 ИЗУЧАЕМАЯ ТЕМА «ДИАГРАММА АВТОМАТА»

Практическое задание по данной теме выполняется с использованием [2]. Изучается материал тринадцатой главы, стр. 130 – 162 (4.2. Диаграммы автомата).

#### 3.1 Ассоциированные понятия

Англоязычное название – State Machine Diagram. Abbreviated form – stm.

Автомат (state machine) – совокупность состояний и переходов между ними.

Состояния: простые (simple), составные (composite), ссылочные (submachine), специальные (pseudo).

Переходы: простые и составные.

Надписи над стрелками: событие перехода (trigger event), сторожевое условие (guard condition), действие на переходе (effect).

#### 3.2 Руководящие принципы построения диаграммы автомата

##### 3.2.1 Общие замечания

Как и любая другая диаграмма, рассматриваемая диаграмма автомата является не самостоятельно значимым элементом проекта, а выражающим некоторое содержательное проектное решение, – определяющее поведение (алгоритм / сценарий функционирования) некоторой части проектируемой системы в процессе реализации конкретного прецедента. Другими словами, диаграмма автомата – это альтернативная форма представления алгоритмической реализации сценариев.

Название реализуемого алгоритма / сценария следует использовать в подрисуночной подписи к диаграмме автомата, как это представлено на последующих рисунках.

Разработку диаграмм автомата следует осуществлять поакторно – вначале реализуется *сценарий использования* проектируемой системы первым актором, за тем со вторым и т.д.

NB

В рамках реализации сценария реализуются варианты использования, ассоциированные с соответствующим актором.

### **3.2.2 Принцип иерархии / декомпозиции в диаграммах автомата**

Проявление принципа иерархии / декомпозиции в диаграммах автомата полностью аналогично проявлению этого принципа в диаграммах прецедентов и деятельности:

1) вначале изображается укрупненная версия сценария использования системы актором; если на диаграмме вариантов использования актор ассоциирован с несколькими вариантами, то сценарий поведения актора должен включать процесс выбора актором необходимого варианта использования системы; как правило, указанный процесс не отображается на диаграмме в виде самостоятельного прецедента, – он подразумевается, носит чисто «технический» характер и примитивен в своей реализации; его отсутствие позволяет сосредоточить внимание читателя на ассоциациях актора именно с вариантами использования системы, не отвлекая внимания на технический процесс выбора варианта; иллюстрирующий пример показан на рисунке 6;

2) затем осуществляется алгоритмическая реализация вариантов использования в виде диаграмм автоматов; при этом вспомогательные прецеденты могут представляться и виде деятельностей, как это показано на рисунке 7, и в виде состояний, как это показано на рисунках 8 и 9;

3) далее осуществляется декомпозиция – алгоритмическая реализация вспомогательных прецедентов, если в этом имеется необходимость;

4) деятельностей состояний, чьи названия не исчерпывают семантику выполняемых операций, так же подлежат алгоритмической реализации;

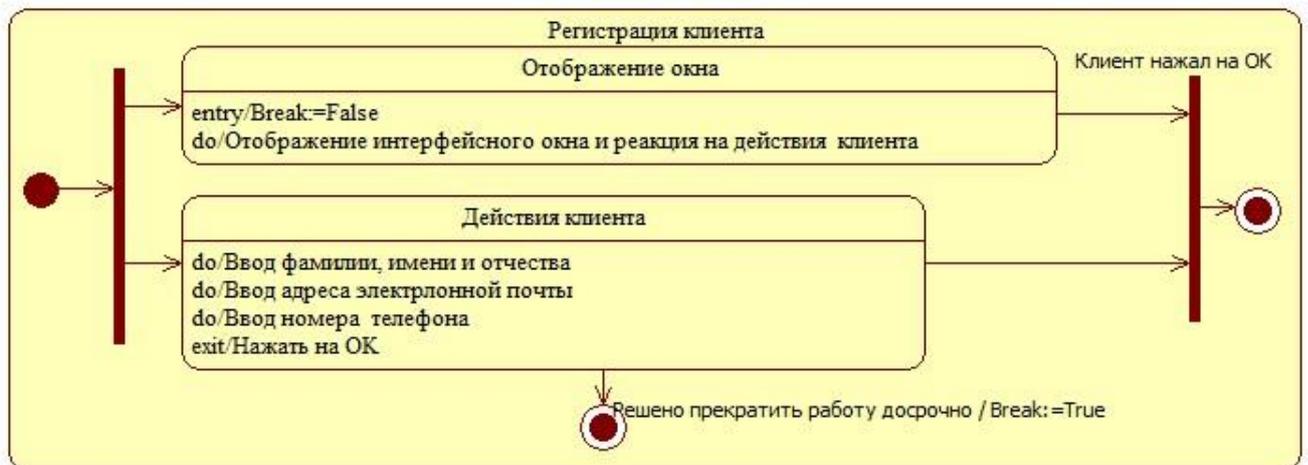
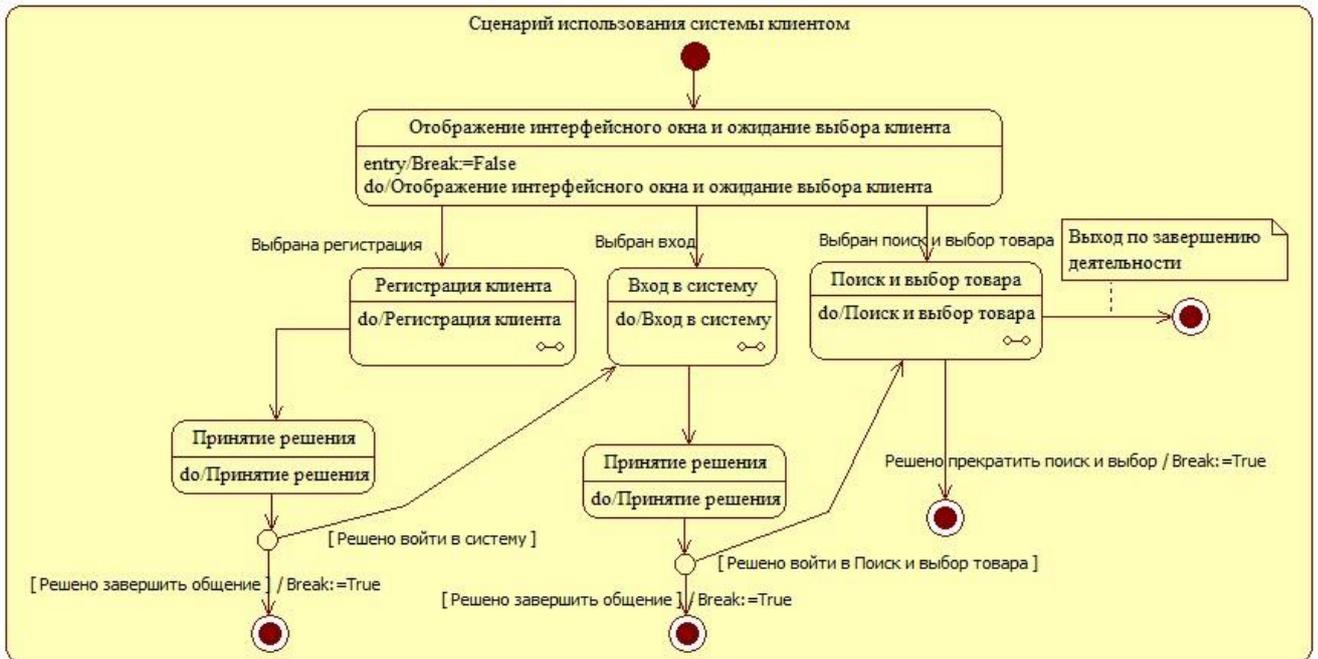
5) для примитивных деятельностей указывается факт отсутствия их алгоритмических реализаций по причине их семантической простоты;

6) процесс декомпозиции может быть остановлен в случае существенного превышения рационального объёма отчётных материалов.

### **3.2.3 Примеры диаграмм автомата**

Диаграмма, представленная на рисунке 6, – это пример алгоритмической реализации сценария работы клиента с проектируемой системой. Сценарий представлен в виде последовательного составного состояния – содержащего в себе один автомат. Существенной особенностью этого автомата является то, что состояние «Стартовое меню» содержит внутреннюю деятельность «Отображение интерфейсного окна ожидание выбора клиента» с потенциально бесконечной длительностью, – клиент может медлить с выбором теоретически бесконечно, поэтому выход из этого состояния возможен только по триггерным событиям. Таких события три – по числу возможных вариантов использования.

Сами варианты использования представлены простыми состояниями



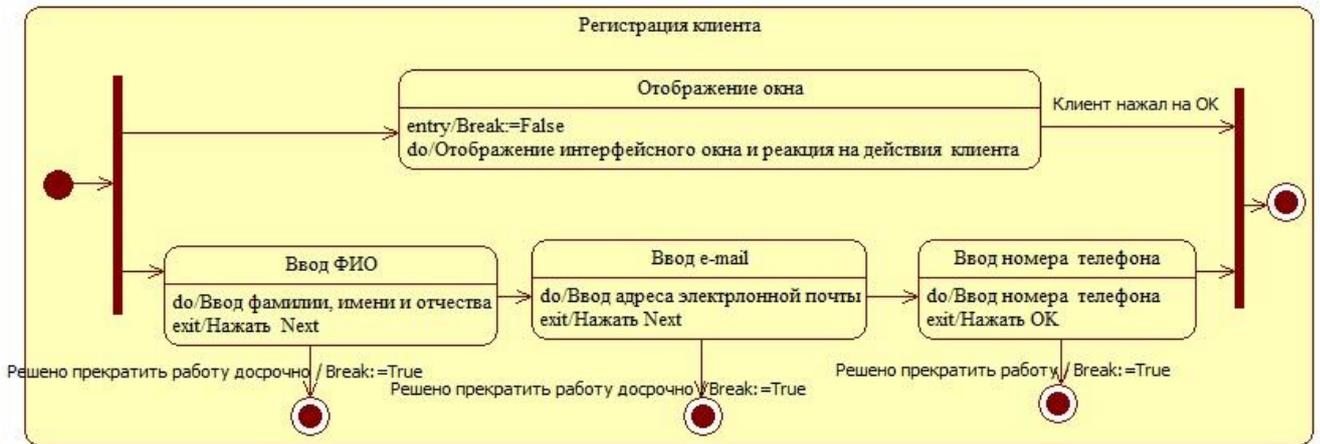


Рисунок 7 – Алгоритм реализации прецедента «Регистрация клиента» в форме автомата. Версия 2

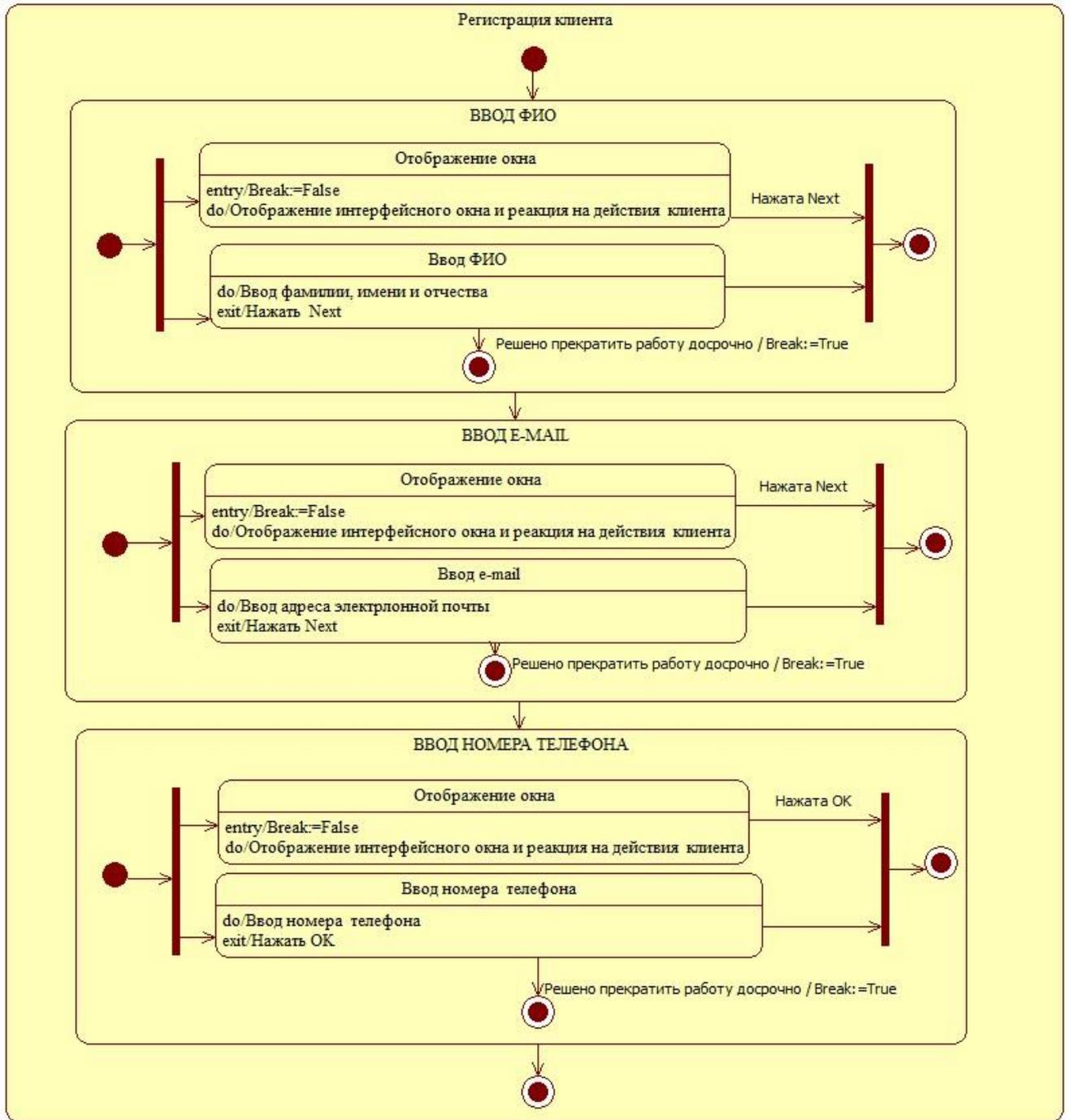


Рисунок 7 – Алгоритм реализации прецедента «Регистрация клиента» в форме автомата. Версия 3

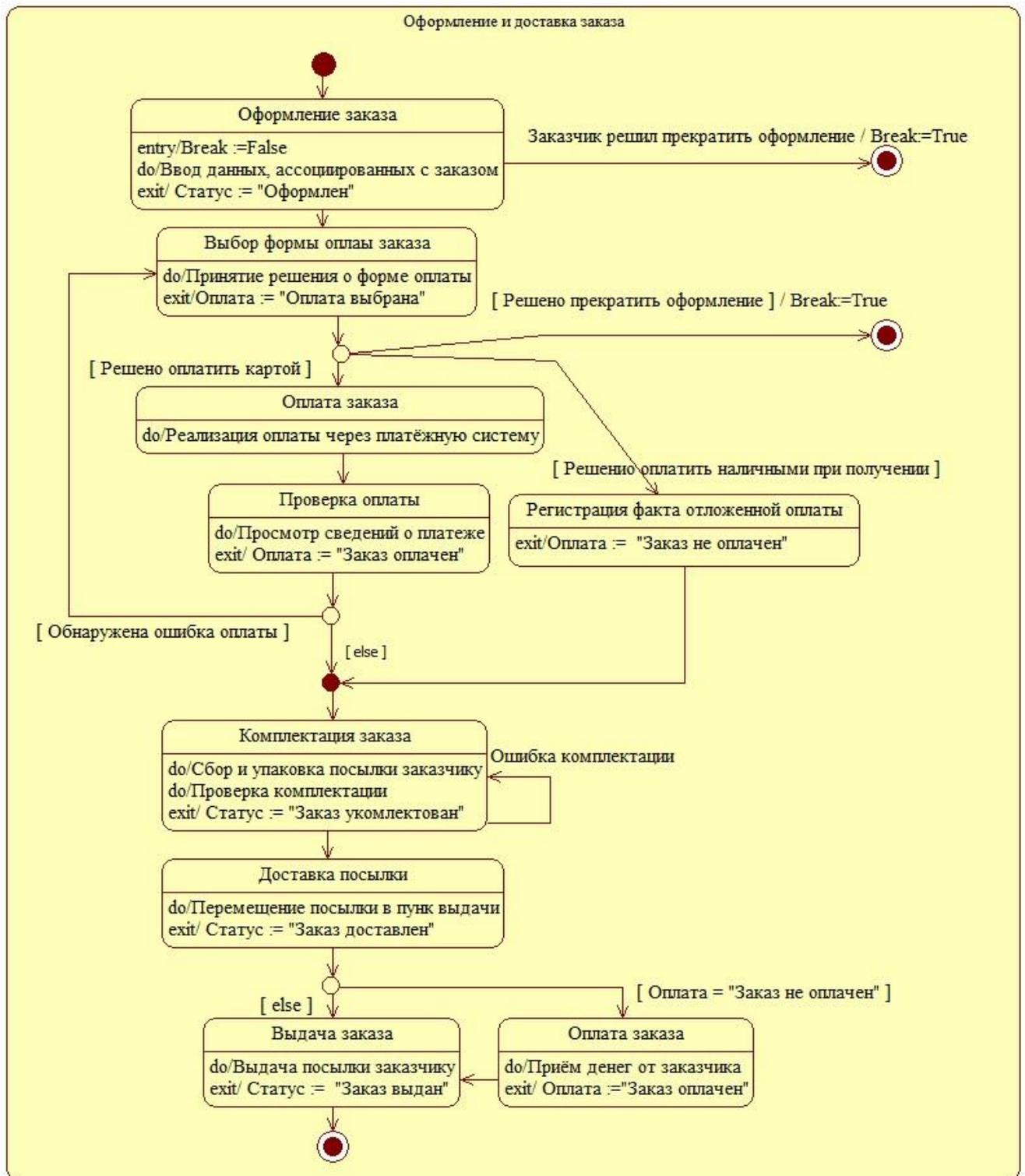


Рисунок 7 – Алгоритм реализации прецедента «Оформление и доставка заказа» в форме автомата

## ЛИТЕРАТУРА

1. Каюмова А.В. Визуальное моделирование систем в StarUML: Учебное пособие/ А.В. Каюмова. Казань. – Казанский федеральный университет, 2013. – 104 с.

2. Иванов, Денис Юрьевич. Унифицированный язык моделирования UML [Электронный ресурс]: учебное пособие для вузов по направлению подготовки "Системный анализ и управление" / Д.Ю. Иванов, Ф.А. Новиков; Санкт-Петербургский государственный политехн. ун-т. – Электрон. текстовые дан. (1 файл: 1,83 Мб). – Санкт-Петербург, 2011. – Загл. с титул. экрана. – Электронная версия печатной публикации. – Свободный доступ из сети Интернет (чтение, печать, копирование). – Текстовый документ. – Adobe Acrobat Reader 7.0. Доступно по URL:<http://elib.spbstu.ru/dl/2962.pdf>, <http://elib.spbstu.ru/dl/2962.pdf/download/2962.pdf>

3. OMG Unified Modeling Language TM (OMG UML). Version 2.5 // OMG Document Number formal/2015-03-01. Normative Reference: <http://www.omg.org/spec/UML/2.5>. – 794 pp.