

ПЕРМСКИЙ
ГОСУДАРСТВЕННЫЙ
НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Н. Н. Дацун

МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

ПРАКТИКУМ



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«ПЕРМСКИЙ ГОСУДАРСТВЕННЫЙ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»

Н. Н. Дацун

МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

ПРАКТИКУМ

*Допущено методическим советом
Пермского государственного национального
исследовательского университета в качестве
учебно-методического пособия для студентов,
обучающихся по направлению подготовки бакалавров
«Прикладная математика и информатика»*



Пермь 2022

УДК 004.434: 004.94

ББК 32.97

Д215

Дацун Н. Н.

Д215 Моделирование информационных систем. Практикум [Электронный ресурс]: учебно-методическое пособие / Н. Н. Дацун ; Пермский государственный национальный исследовательский университет. – Электронные данные. – Пермь, 2022. – 6,47 Мб ; 127 с. – Режим доступа: <http://www.psu.ru/files/docs/science/books/uchebnie-posobiya/Dacun-Modelirovanie-Informacionnyh-Sistem-Praktikum.pdf>. – Заглавие с экрана.

ISBN 978-5-7944-3841-3

Пособие представляет собой методические рекомендации по выполнению практической части дисциплины «Моделирование информационных систем» при объектно-ориентированном подходе к моделированию. Сформулированы общие методические указания и требования к комплексу работ. Для каждой темы занятий приведен список типичных ошибок, которые допускаются студентами в моделях и в сопровождающих их описаниях. Ошибки, допущенные в диаграммах UML, классифицируются по видам: лексические, синтаксические и семантические, а также ошибки размещения. Примеры ошибок сопровождаются комментариями, объясняющими их возникновение и направленными на устранение и недопущение подобных ошибок.

Издание предназначено для студентов очной формы обучения: бакалавриата направления подготовки «Прикладная математика и информатика» специализации «Системное программирование и компьютерные технологии» и студентов специальности «Компьютерная безопасность» специализации «Разработка защищенного программного обеспечения», изучающих дисциплину «Моделирование информационных систем». Учебное пособие будет полезно студентам других ИТ-направлений и специальностей.

УДК 004.434: 004.94

ББК 32.97

*Издается по решению ученого совета механико-математического факультета
Пермского государственного национального исследовательского университета*

Рецензенты: профессор кафедры «Автоматика и телемеханика» Пермского национального исследовательского политехнического университета, Заслуженный изобретатель РФ, д-р техн. наук, профессор **С. Ф. Тюрин**;

кафедра информационных технологий в бизнесе НИУ ВШЭ – Пермь (зав. кафедрой – д-р пед. наук, профессор **Е. Г. Плотникова**, доцент кафедры – канд. физ.-мат. наук, доцент **Л. Н. Лядова**).

ISBN 978-5-7944-3841-3

© Дацун Н. Н., 2022

© ПГНИУ, 2022

СОДЕРЖАНИЕ

Перечень сокращений и обозначений.....	4
Введение.....	5
1. Создание UCD с низким уровнем детализации.....	7
2. Описание прецедентов.....	15
3. Создание UCD с высоким уровнем детализации.....	21
4. Создание AD.....	34
5. Концептуальное моделирование.....	50
6. Создание диаграмм понятий.....	60
7. Создание SD.....	72
8. Описание системных операций.....	87
9. Создание диаграмм сотрудничества.....	94
10. Создание CD.....	107
11. Создание диаграмм компонентов.....	118
12. Создание диаграмм развертывания студентов.....	123
13. Выписки из учебных планов.....	125
Список использованных источников.....	126

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящем пособии применяют следующие сокращения и обозначения:

AD	– <i>англ.</i> Activity Diagram, диаграмма активностей (деятельности) языка UML
BPMN	– <i>англ.</i> Business Process Management Notation, язык моделирования бизнес-процессов
CollD	– <i>англ.</i> Collaboration diagram, диаграмма сотрудничества (коммуникации, кооперации) языка UML
CompD	<i>англ.</i> Component Diagrams, диаграмма компонентов языка UML
CD	– <i>англ.</i> Class Diagram, диаграмма классов языка UML
DD	<i>англ.</i> Deployment Diagram, диаграмма развертывания языка UML
Hight LoD	– <i>англ.</i> High Level of Detail, высокий уровень детализации
LoD	– <i>англ.</i> Level of Detail, уровень детализации
Low LoD	– <i>англ.</i> Low Level of Detail, низкий уровень детализации
MSC	– <i>англ.</i> Message sequence chart, диаграмма последовательности сообщений
SD	– <i>англ.</i> Sequence diagram, диаграмма последовательностей
UCD	– <i>англ.</i> Use Case Diagram, диаграмма прецедентов (вариантов использования) языка UML
UML	– <i>англ.</i> Unified Modeling Language, унифицированный язык моделирования
англ	– английский
БД	– база данных
ГОСТ	– межгосударственный стандарт
ГОСТ Р	– национальный стандарт Российской Федерации
др.	– другие
ИС	– информационная система
напр.	– например
ООАП	– объектно-ориентированный анализ и проектирование
ООП	– объектно-ориентированное программирование
МИС	– моделирование информационных систем
ПГНИУ	– Пермский государственный национальный исследовательский университет
ПрО	– предметная область
СО	– системная операция
СС	– системное сообщение
СУБД	– система управления базой данных
т.д.	– так далее
т.к.	– так как
т. п.	– тому подобное
т.ч.	– том числе

ВВЕДЕНИЕ

В базовой части учебных планов направления подготовки «Прикладная математика и информатика» (специализация «Системное программирование и компьютерные технологии») очной формы обучения и в вариативной части (элективы «Профессиональные») учебных планов специальности «Компьютерная безопасность» (специализация «Разработка защищенного программного обеспечения») предусмотрено изучение дисциплины «Моделирование информационных систем». Целью этой дисциплины является подготовка студентов, способных анализировать задачи, стоящие перед ИС, выполнять их моделирование, документировать и реализовывать их на различных этапах жизненного цикла ИС. Практический блок дисциплины «МИС» направлен на приобретение практических навыков применения методологий моделирования, проектирования и реализации информационных систем для решения прикладных задач, на выработку на основе приобретенных знаний и навыков необходимых компетенций для решения конкретных профессиональных задач.

Комплекс лабораторных работ, практических занятий и самостоятельной работы студентов по дисциплине «Моделирование информационных систем» решает задачи формирования следующих компетенций: способность использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанных с математическими и компьютерными науками; анализировать социально значимые проблемы и процессы; создавать, анализировать, реализовывать математические и информационные модели с применением современных вычислительных систем; применять языки, системы и инструментальные средства программирования, работать с программными средствами прикладного, системного и специального назначения в профессиональной деятельности; способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения; владеть базовыми знаниями в области информатики, навыками использования программных средств и работы в компьютерных сетях, приобретать новые знания, используя современные информационные технологии; умение разрабатывать и оформлять различную проектную и техническую документацию и др.

В пособии представлены сведения о типичных ошибках, которые допускают студенты при создании моделей и их описании на таких этапах жизненного цикла ИС, как анализ, проектирование и реализация:

- создание UCD с низким LoD,
- описание прецедентов,

- создание UCD с высоким LoD,
- создание AD,
- концептуальное моделирование,
- создание CD с низким LoD,
- создание SD,
- описание CO,
- создание CollD,
- создание CD с высоким LoD,
- создание CompD,
- создание DD.

Пособие содержит список печатных изданий и электронных источников.

Текст методического пособия подготовлен в соответствии с ГОСТ 7.32 [1] и ГОСТ Р 7.0.5 [2].

1. Создание UCD с низким уровнем детализации

В моделировании на языке UML [3-6] уровень детализации (LoD) характеризует объем информации, которая используется для представления элемента моделирования: чем больше объем информации, который используется для представления моделируемой информации, помещенной в модель, тем больше известно об элементе. Поэтому различают модели с High LoD и с Low LoD [7, 8].

Когда LoD, используемый в модели UML, низкий, то:

а) для понятий или знаний, описанных в модели, используются только несколько синтаксических функций, таких как имя субъекта (прецедента в диаграмме прецедентов, объекта в диаграммах активностей, последовательностей, взаимодействия, класса в диаграммах классов и т. п.),

б) нет существенной разницы в ассоциациях субъектов без указания каких-либо дополнительных фактов об этих субъектах,

в) отсутствуют названия ассоциаций, их направленность и пр.

Результат формулирования функциональных требований к ИС при объектно-ориентированном подходе к моделированию необходимо сначала представить в виде диаграмм прецедентов с низким уровнем детализации. Это является целью темы 1. При этом в UCD должны быть использованы только такие субъекты, как система, акторы, прецеденты и ассоциации между акторами и прецедентами.

Ошибки, допускаемые студентами при создании диаграмм прецедентов, классифицируются по аналогии с видами ошибок в программном коде [9, 10]: лексические, синтаксические и семантические [11]. Также допускаются ошибки размещения элементов диаграмм [11], связанные с тем, что UML – это графический язык моделирования.

Лексическими являются ошибки, которые нарушают правила создания и использования лексем подмножества UML для создания диаграмм прецедентов с низким LoD – допустимых графических образов субъектов (система, акторы, прецеденты и ассоциации) и их именования.

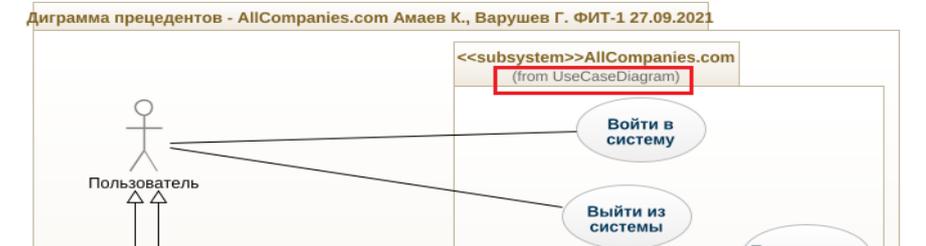
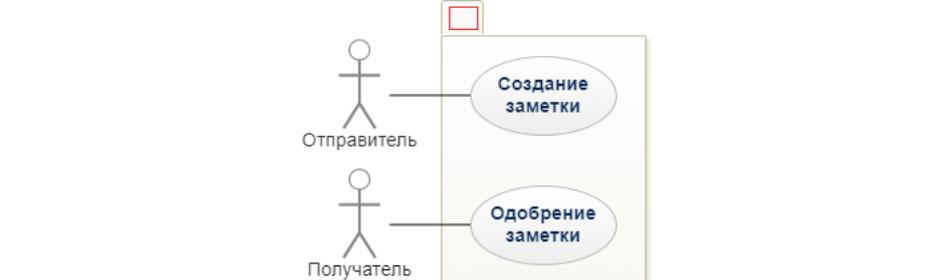
Синтаксические ошибки возникают при отсутствии субъектов, неверном установлении связей между субъектами и/или их размещении по отношению друг к другу, а также при нарушении принципа идентификации.

К семантическим ошибкам приводит неверное понимание уровня моделирования («скатывание» с уровня модели на уровень данных) или неверное выделение прецедентов и ассоциированных с ними акторов.

Для создания диаграмм прецедентов следует использовать сервис genMyModel [12].

В таблице 1.1 представлены типичные ошибки, допускаемые студентами при создании UCD с низким уровнем детализации.

Таблица 1.1 – Ошибки в диаграммах прецедентов с низким уровнем детализации

Номер и текст ошибки	Пример с ошибкой	
1	2	
Лексические ошибки		
1.1 Использование элементов, которые не являются лексемами для UCD		<p>Диаграмма прецедентов - StateCompany,</p> 
Элементы, выделенные в моделях, не входят в список, разрешенный стандартом UML для UCD		
1.2 Отсутствие границы системы		
Для создания границы системы следует использовать элемент, который показан на рисунке модели красным контуром		
1.3 Название системы неверное	<p>Диаграмма прецедентов - AllCompanies.com Амаев К., Варушев Г. ФИТ-1 27.09.2021</p> 	
Название системы не должно содержать символы, недопустимые в идентификаторах (скобки и символы операций)		
1.4 Отсутствие названия системы (имя субъекта)		
Название системы (краткое) было определено в концепции ИС. Его необходимо использовать для идентификации системы и идентификации модели (напр., Notes)		

Продолжение таблицы 1.1

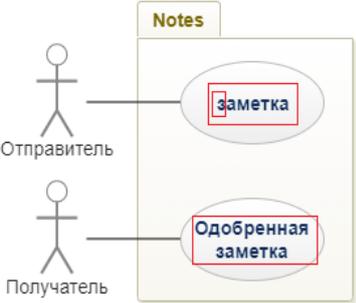
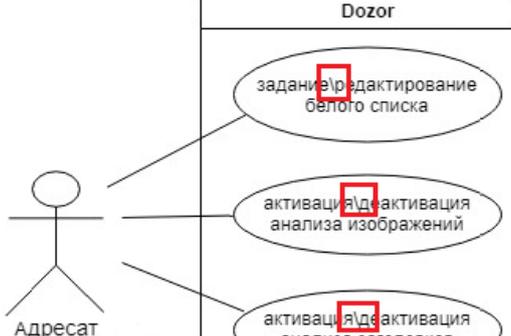
1	2
<p>1.5 Акторы должны быть представлены соответствующим элементом UCD</p>	

Использован неверный символ актора. GenMyModel не позволит создать такие элементы при использовании панели «Use case diagram»

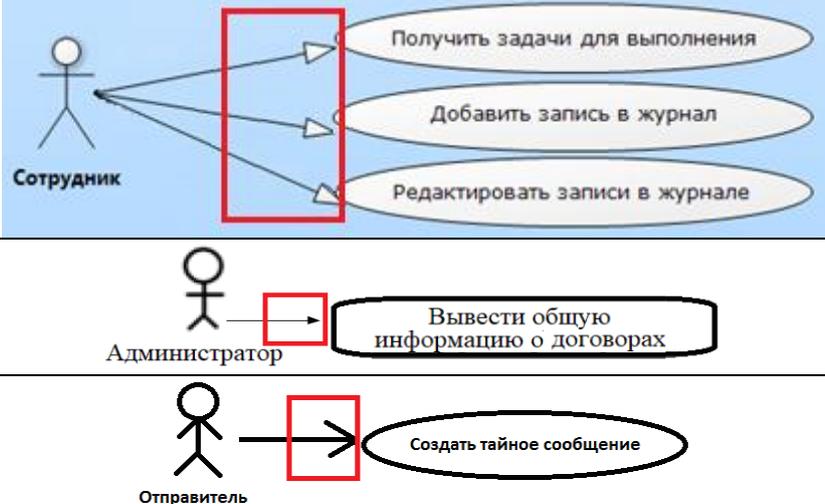
<p>1.6 Неверное имя актора</p>		

Имя актора должно быть представлено именем существительным, начинаться с заглавной буквы и не содержать символы, недопустимые в идентификаторах. Лишними являются слова «Актор» и контур прямоугольника вокруг имени. Допустима нотация как в диаграммах последовательностей для обозначения объектов (прямоугольник с именем), но без пиктограммы

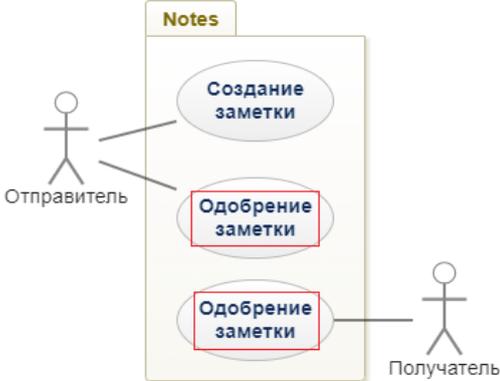
Продолжение таблицы 1.1

1	2	
<p>1.7 Прецеденты должны быть представлены соответствующим элементом UCD</p>		
<p>Неверный символ прецедента. GenMyModel не позволит создать такие элементы при использовании панели «Use case diagram»</p>		
<p>1.8 Неверное имя прецедента</p>		
		
<p>Имя прецедента должно быть представлено в виде действия (глаголом в инфинитиве или именем существительным отглагольной формы), начинаться с заглавной буквы и не содержать символы, недопустимые в идентификаторах</p>		
<p>1.9 Перемешивание в именах прецедентов глаголов и имен существительных</p>		
<p>Следует придерживаться во всех вариантах использования единого стиля написания имен: или глаголы, или существительные</p>		

Продолжение таблицы 1.1

1	2
<p>1.10 Неверный символ ассоциации</p>	
<p>GenMyModel не позволит создать такие элементы при использовании панели «Use case diagram» между актором и прецедентом</p>	
<p>Синтаксические ошибки</p>	
<p>1.11 Размещение названия системы внутри нее</p>	
<p>Также допущена лексическая ошибка в названии системы: лишним является слово «граница»</p>	
<p>1.12 Размещение актора внутри системы</p>	
<p>Актор в UML – это внешний по отношению к системе объект (человек или другая система). Поэтому актор всегда должен быть размещен за границей системы</p>	

Продолжение таблицы 1.1

1	2
<p>1.13 Система как актер</p>	<p>Диаграмма прецедентов. Осов А.А. гр. ПМИ-3,4-2017, 18.01.2020</p> 
<p>Актер представляет собой внешнюю среду по отношению к системе. Поэтому система не может быть внешней средой по отношению к самой себе</p>	
<p>1.14 Размещение прецедента вне системы</p>	
<p>Прецедент в UML – это описание множества содержательно близких сценариев взаимодействия акторов с ИС. Поэтому прецедент всегда должен быть размещен внутри границы системы</p>	
<p>1.15 Имена акторов должны быть уникальными</p>	
<p>Нарушение принципа идентификации. Обратите внимание на имена акторов при объединении нескольких диаграмм прецедентов: в исходных UCD имена акторов могут быть уникальными, но в объединенной диаграмме это свойство может быть утеряно</p>	
<p>1.16 Имена прецедентов должны быть уникальными</p>	
<p>Нарушение принципа идентификации. Обратите внимание на имена прецедентов при объединении нескольких диаграмм прецедентов: в исходных UCD имена прецедентов могут быть уникальными, но в объединенной диаграмме это свойство может быть утеряно</p>	

Продолжение таблицы 1.1

1	2
<p>1.17 Прецедент не имеет отношений с другими элементами диаграммы</p>	
<p>Прецедент в UML описывает множество содержательно близких сценариев взаимодействия акторов с ИС. Поэтому с прецедентом всегда должен быть ассоциирован хотя бы один актер</p>	
<p>1.18 Актер не имеет ни одного отношения ассоциации с прецедентами</p>	
<p>Цель актора – это получение некоторого полезного результата при помощи системы. Поэтому с актором всегда должен быть ассоциирован хотя бы один прецедент</p>	
<p>Семантические ошибки</p>	
<p>1.19 Лишний актер</p>	
<p>1.19 Лишний актер</p>	
<p>Сущности, которые не запрашивают систему предъявить свою функциональность и не получают некоторый полезный результат при помощи системы, актерами не являются</p>	

Продолжение таблицы 1.1

1	2
<p>1.20 Лишний пре- цедент</p>	
<p>При функционировании объектно-ориентированной системы все ее объекты размещены в оперативной памяти. Изменение значений соответствующих атрибутов отдельного объекта представляет собой изменение его состояния и состояния системы в целом. Процесс «Сохранение» относится к уровню данных, но не к уровню модели</p>	
<p>1.21 Два и более актера с одина- ковыми набора- ми ассоцииро- ванных преце- дентов</p>	
<p>Это ошибка-предупреждение: такой сценарий взаимодействия с системой говорит о том, что должен быть один актер, ассоциированный с такими прецедентами. Уточнение и вариант исправления прецедента рассмотрены в теме 3 (таблица 3.1)</p>	

2. Описание прецедентов

Следует помнить, что у вариантов использования после присвоения им имен при первом упоминании в списке функциональных требований эти имена должны быть одинаковыми далее:

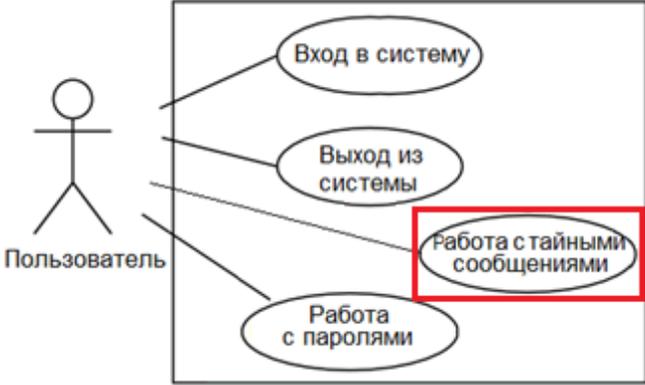
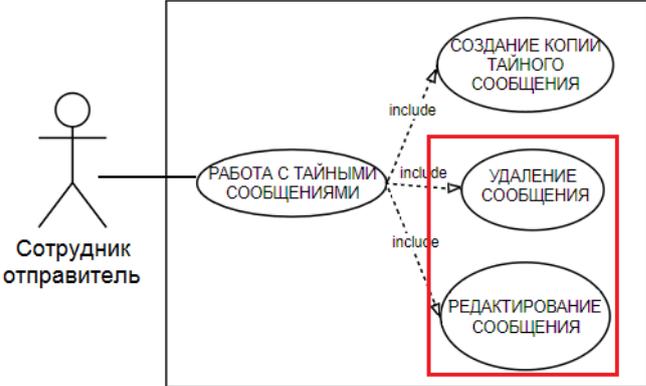
- в UCD с низким и высоким уровнем детализации,
- в классификации прецедентов,
- в высокоуровневом и развернутом прецедентах.

В таблице 2.1 представлены типичные ошибки, допускаемые студентами при описании функциональных требований в виде прецедентов.

Таблица 2.1 – Ошибки в описании функциональных требований в виде прецедентов и их классификации

Номер и текст ошибки	Пример с ошибкой		
1	2		
2.1 Неверное имя актора	<p>3.1 Описание функциональных требований Для кассира можно выделить варианты использования:</p> <ul style="list-style-type: none"> – просмотр транзакции счёта; – редактирование транзакции счёта... 		
	<p><u>Название:</u> Просмотр транзакции счёта <u>Акторы:</u> кассир <u>Описание:</u> кассир ищет счёт по уникальному полю. Просматривает изменения в транзакции по выбранному счёту.</p>		
Лексика имени актора не изменяется, она единая на всех этапах и шагах ООАП и реализации			
2.2 Не используются имена акторов	<p><u>Название:</u> Просмотр заметки <u>Акторы:</u> Отправитель, Получатель <u>Описание:</u> Открытие и чтение заметки, не имеющей отметки о прочтении.</p>		
В высокоуровневом прецеденте список акторов содержит два имени. В поле «Описание» имена этих акторов не используются			
2.3 Нарушен синтаксис описания прецедентов	<p><u>Название:</u> Работать с полученной заметкой <u>Акторы:</u> Получатель <u>Краткое описание:</u> Получатель начинает работу с полученной от Отправителя заметкой. <u>Триггер:</u> Получатель выбирает заметку из списка актуальных заметок. <u>Основной поток:</u></p>		
	<table border="1"> <tr> <td>Получатель инициирует выбор заметки из списка.</td> <td>Система предоставляет получателю информацию о заметке.</td> </tr> </table>	Получатель инициирует выбор заметки из списка.	Система предоставляет получателю информацию о заметке.
	Получатель инициирует выбор заметки из списка.	Система предоставляет получателю информацию о заметке.	
	<table border="1"> <tr> <td>Получатель работает с информацией о заметке (подпоток R1), отмечает заметку прочитанной.</td> <td>Система фиксирует информацию о том, что получатель прочитал заметку, оставляет на ней соответствующую метку о прочтении.</td> </tr> </table>	Получатель работает с информацией о заметке (подпоток R1), отмечает заметку прочитанной.	Система фиксирует информацию о том, что получатель прочитал заметку, оставляет на ней соответствующую метку о прочтении.
Получатель работает с информацией о заметке (подпоток R1), отмечает заметку прочитанной.	Система фиксирует информацию о том, что получатель прочитал заметку, оставляет на ней соответствующую метку о прочтении.		
<p><u>Подпоток:</u> <u>R1:</u> «Прочитать заметку»: Получатель «сканирует» информацию о заметке отправителя, предоставленной Системой.</p>			

Продолжение таблицы 2.1

1	2
<p>Отсутствуют подписи у основного потока: столбцы должны именоваться «Действия акторов» и «Отклик системы». Отсутствует нумерация действий в столбцах у основного потока (1, 2, ...). Обозначения подпотоков рекомендовано начинать с символа 'S' (от англ. «subflow»). В названии подпотока кавычки не нужны</p>	
<p>2.4 Несогласованность имен прецедентов в списке функциональных требований и в UCD с низким уровнем детализации</p>	<p>3.1 Описание функциональных требований Исходя из потребностей пользователя, выделяются варианты использования:</p> <ul style="list-style-type: none"> – вход в систему, – выход из системы, – создание копии тайного сообщения, – редактирование тайного сообщения, – удаление тайного сообщения... 
<p>У варианта использования должно быть одинаковое имя при его первом упоминании в списке функциональных требований и в диаграмме прецедентов с High LoD. Варианты таких ошибок: в UCD отсутствуют прецеденты с именами из описания функциональных требований или в описании функциональных требований нет прецедентов с именами, использованными в UCD</p>	
<p>2.5 Несогласованность имен прецедентов в списке функциональных требований и в UCD с высоким уровнем детализации</p>	<p>3.1 Описание функциональных требований Исходя из потребностей пользователя, выделяются варианты использования:</p> <ul style="list-style-type: none"> – создание копии тайного сообщения, – редактирование тайного сообщения, – удаление тайного сообщения... 
<p>У прецедентов должны быть одинаковые имена в списке функциональных требований и в UCD с High LoD</p>	

Продолжение таблицы 2.1

1	2									
<p>2.6 Несогласованность имен прецедентов в списке функциональных требований и в классификации прецедентов</p>	<p>3 Описание функциональных требований Исходя из потребностей администратора, выделяем варианты использования:</p> <ul style="list-style-type: none"> – создание счёта; – просмотр списка счетов; – редактирование отдельных записей счёта. 	<p>4 Классификация прецедентов Главный: создание нового счёта. Второстепенный: редактирование счёта. ...</p>								
<p>У вариантов использования должны быть одинаковые имена в списке функциональных требований и в их классификации</p>										
<p>2.7 Несогласованность имен прецедентов в списке функциональных требований и в высокоуровневом прецеденте</p>	<p>3 Описание функциональных требований Исходя из потребностей администратора, выделяем варианты использования:</p> <ul style="list-style-type: none"> – просмотр списка счетов; – редактирование отдельных записей счёта <p><u>Название:</u> Просмотр списка счетов и их редактирование <u>Акторы:</u> Администратор <u>Описание:</u> Администратор ищет в ИС существующий или создает новый счёт. В новом счете указывает его название, наименование банка, номер счета в банке. Если счет уже есть в ИС, Администратор может просмотреть данные о счёте или внести в него изменения.</p>									
<p>У прецедентов должны быть одинаковые имена в списке функциональных требований и в их высокоуровневом описании</p>										
<p>2.8 Несогласованность имен прецедентов в списке функциональных требований и в развернутом прецеденте</p>	<p>3 Описание функциональных требований Для роли кассира можно выделить варианты использования:</p> <ul style="list-style-type: none"> – просмотр транзакций по выбранному счету; – редактирование транзакций по выбранному счёту. <p><u>Название:</u> просмотр списка транзакций по выбранному счёту и их редактирование <u>Акторы:</u> Кассир <u>Краткое описание:</u> Кассир ищет в ИС счёт по его номеру. Просматривает или вносит изменения в транзакцию по выбранному счёту и сохраняет её... <u>Основной поток:</u></p> <table border="1" data-bbox="432 1585 1425 1861"> <thead> <tr> <th data-bbox="432 1585 948 1630">Действия акторов</th> <th data-bbox="948 1585 1425 1630">Отклик системы</th> </tr> </thead> <tbody> <tr> <td data-bbox="432 1630 948 1706">1. Кассир указывает номер счёта.</td> <td data-bbox="948 1630 1425 1706">2. Система предоставляет данные найденного счёта.</td> </tr> <tr> <td data-bbox="432 1706 948 1816">3. Кассир просматривает транзакции счёта и выбирает одну для редактирования.</td> <td data-bbox="948 1706 1425 1816">4. Система предоставляет данные выбранной транзакции.</td> </tr> <tr> <td data-bbox="432 1816 948 1861">5. ...</td> <td data-bbox="948 1816 1425 1861"></td> </tr> </tbody> </table>		Действия акторов	Отклик системы	1. Кассир указывает номер счёта.	2. Система предоставляет данные найденного счёта.	3. Кассир просматривает транзакции счёта и выбирает одну для редактирования.	4. Система предоставляет данные выбранной транзакции.	5. ...	
Действия акторов	Отклик системы									
1. Кассир указывает номер счёта.	2. Система предоставляет данные найденного счёта.									
3. Кассир просматривает транзакции счёта и выбирает одну для редактирования.	4. Система предоставляет данные выбранной транзакции.									
5. ...										
<p>У прецедентов должны быть одинаковые имена в списке функциональных требований и в их развернутом описании</p>										

Продолжение таблицы 2.1

1	2	
<p>2.9 Подмена действия «уничтожение» действием «удаление»</p>	<p><u>Название:</u> Удаление записи <u>Актеры:</u> Редактор <u>Описание:</u> Редактор может удалять запись. Изменение не будет принято, пока администратор не подтвердит изменение. ...</p>	
	<p><u>Название:</u> Работа с заданием <u>Актеры:</u> Администратор <u>Краткое описание:</u> Администратор работает с выбранным заданием из проекта. <u>Триггер:</u> Начало работы с заданием. <u>Основной поток:</u></p>	
	<p>Действия акторов</p>	<p>Отклик системы</p>
	<p>1. Администратор взаимодействует с заданием, при этом выполняются подпотоки S1-S3.</p>	<p>2. Система предоставляет пользователю выбор действий над заданием.</p>
	<p><u>Подпотоки:</u> S1: Создание нового задания. Администратор создает новое задание, указывая свойства задания. S2: Модификация существующего задания. Администратор редактирует выбранные свойства существующего задания. S3: Удаление существующего задания. Администратор удаляет существующее задание. Задание также удаляется у сотрудников, которые с ним были связаны.</p>	
<p>Удаление сущности может производиться из множества сущностей, «удаление» не есть действие с сущностью, это действие с множеством сущностей</p>		
<p>2.10 Неверная классификация прецедентов</p>	<p>Главные: Вход в систему Ввод пароля к тайным сообщениям Второстепенные: Чтение тайных сообщений Отправка тайных сообщений Вспомогательные: Управление собственной архивной копией сообщений Управление основной копией секретных сообщений</p>	
<p>Если работа с тайными сообщениями («Отправка...» и «Чтение...») отнесена ко второстепенным прецедентам, то автор такой классификации считает, что без их реализации возможно частичное функционирование системы. Но тогда функциями такой ИС являются только «Вход в систему» и «Ввод пароля к тайным сообщениям», без реализации которых невозможно функционирование системы. Поэтому приведенная классификация ошибочная: прецеденты «Чтение тайных сообщений» и «Отправка тайных сообщений» следует классифицировать как главные и в дальнейшем для них построить остальные модели проекта</p>		

Продолжение таблицы 2.1

1	2	
<p>2.11 У подпотоков не указано место их возникновения в основных потоках</p>	<p><u>Название:</u> Оплата услуг портала. <u>Акторы:</u> Представитель предприятия. <u>Краткое описание:</u> Представитель предприятия производит оплату за выбранный им пакет. За тестовый пакет оплата не взимается. <u>Триггер:</u> Представитель предприятия производит оплату за выбранный им пакет. <u>Основной поток:</u></p>	
	Действия акторов	Отклик системы
	1. Представитель предприятия выбирает пакет услуг.	2. Система предъявляет цену за пакет услуг.
	3. Представитель предприятия оплачивает пакет услуг.	4. Система предоставляет оплаченный пакет услуг.
	<p><u>Подпотоки:</u> <u>S1:</u> Продление текущего пакета. Если для учетной записи уже действует определенный пакет услуг (исключая тестовый), представитель предприятия оплачивает его продление. Прецедент завершается. <u>S2:</u> Выбор стандартного пакета. Представитель предприятия оплачивает стандартный пакет услуг. Прецедент завершается. <u>S3:</u> Выбор VIP-пакета. Представитель предприятия оплачивает VIP-пакет услуг. Прецедент завершается.</p>	
<p>При описании основного потока места возникновения подпотоков указываются его именем Si</p>		
<p>2.12 У альтернативных потоков не указано место их возникновения в основных потоках</p>	<p><u>Название:</u> Редактирование объявления <u>Акторы:</u> Представитель предприятия <u>Краткое описание:</u> Представитель предприятия редактирует объявление с информацией о данном предприятии. После редактирования объявление станет доступно для посетителей только после подтверждения изменений администратором портала. <u>Триггер:</u> Представитель предприятия редактирует объявление с информацией о данном предприятии. <u>Основной поток:</u></p>	
	Действия акторов	Отклик системы
	1. Представитель предприятия редактирует данные с информацией о предприятии.	2. Система отправляет информацию об изменениях администратору портала.
<p><u>Альтернативные потоки:</u> <u>E1:</u> Просмотр подтверждения или отклонения изменений. Представитель предприятия проверяет статус изменений – подтверждены, отклонены или на проверке. В последнем случае прецедент продолжается, в остальных — завершается.</p>		
<p>При описании основного потока места возможного возникновения исключительных ситуаций помечают именем Ei в скобках</p>		

Продолжение таблицы 2.1

1	2	
<p>2.13 Альтернативные потоки не могут возникать на стороне системы</p>	<p><u>Название:</u> Проверка письма на наличие спама <u>Актеры:</u> Email <u>Описание:</u> Система Email проверяет каждое поступившее письмо на наличие признаков спама. Это может быть достигнуто с помощью применения анализа заголовков, изображений, выявления запрещенных слов, превышения баллов фактора спама, обнаружения адреса среди адресов «белого списка». Далее алгоритм системы решает, в какую папку отправить поступившее письмо. <u>Основной поток:</u></p>	
	Действия акторов	Отклик системы
	<p>1. Email получает письмо.</p>	<p>2. Система проверяет письмо на наличие спама по заданным критериям и по полученным баллам отправляет его в нужную папку (E1-E2).</p>
<p><u>Альтернативные потоки:</u> <u>E1:</u> Перемещение письма в папку входящие. Граница фактора спама не превышена. Система отправляет письмо в папку входящие. Прецедент продолжается. <u>E2:</u> Перемещение письма в папку спам. Граница фактора спама превышена. Система отправляет письмо в папку спам. Прецедент завершается.</p>		
<p>При описании основного потока места возможного возникновения исключительных ситуаций помечают именем E_i в скобках</p>		

3. Создание UCD с высоким уровнем детализации

При выполнении темы 3 выполняется переход от диаграмм прецедентов с низким уровнем детализации к диаграммам с высоким LoD. UCD, подготовленные при выявлении функциональных требований к системе, должны быть расширены добавлением таких субъектов, как отношения обобщения (между акторами и между прецедентами), а также зависимостей включения и расширения (между прецедентами). Эти требования являются обязательными, т.к. в предложенных описаниях предметных областей такие связи между акторами и между прецедентами присутствуют. Они должны быть выявлены и описаны в модели функциональных требований на этапе анализа. Отсутствие отношения обобщения между акторами и между прецедентами, а также зависимостей включения и расширения является ошибкой в созданной модели.

Переход от диаграмм прецедентов с Low LoD к диаграммам с High LoD предполагает уточнение и четкое разделение взаимодействия акторов со множеством (набором) каких-то сущностей и отдельной сущностью.

Рисунок 3.1 содержит фрагмент UCD с Low LoD. Прецеденты, представленные на ней, соответствуют тексту описания предметной области. Это результат поверхностного анализа функций ИС. Но взаимодействие акторов со списком задач и отдельной задачей не структурировано. Отсутствует вариант использования, в котором задача рассматривается как элемент списка задач («Добавление задачи в список»).

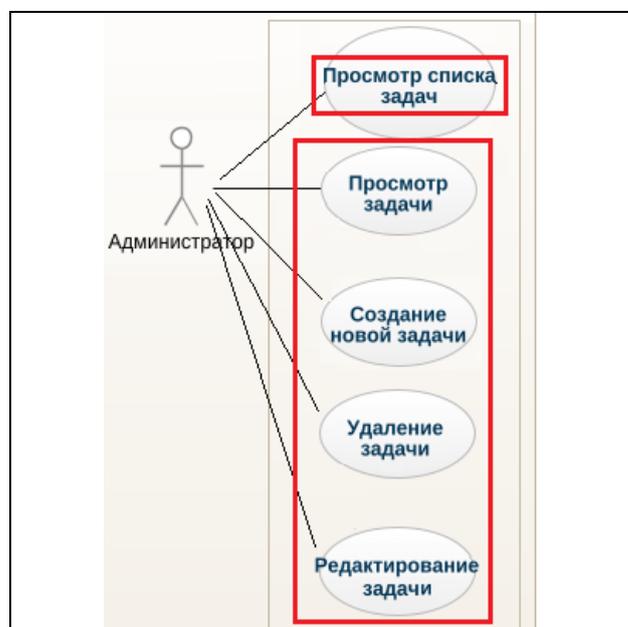


Рисунок 3.1 – Пример UCD с низким уровнем детализации

Кроме этого, в имени прецедента «удаление задачи» отсутствует ссылка на содержащий ее список. Это может свидетельствовать о том, что произошло смешение понятий «Уничтожение задачи» и «Удаление задачи из списка задач», что является семантической ошибкой.

На рисунке 3.2 представлен результат перехода к UCD с High LoD в этой же предметной области.

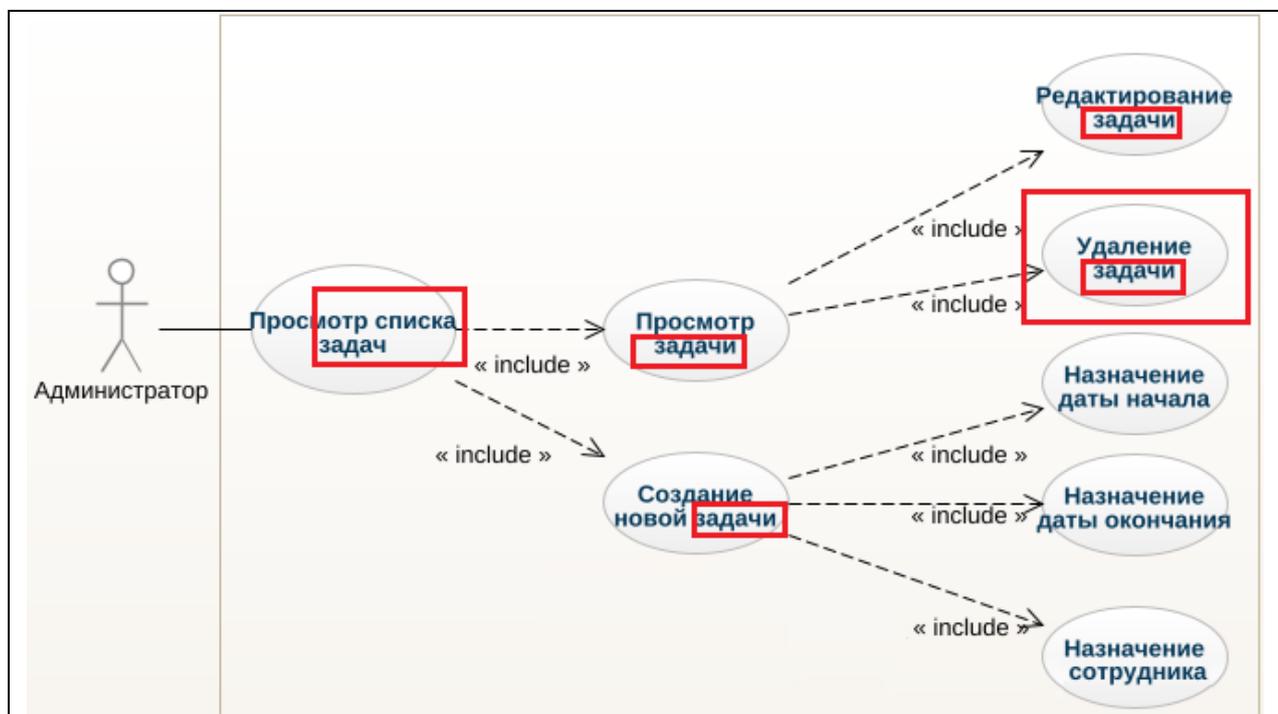


Рисунок 3.2 – Пример перехода к UCD с высоким уровнем детализации

Здесь выполнена попытка структуризации вариантов использования при работе с отдельной задачей. Но явного разделения взаимодействия актора со списком задач и с задачей отсутствует. Отсутствуют сценарии использования задачи как элемента списка задач («Добавление задачи в список...» и «Удаление задачи из списка...»). Прецедент «Удаление задачи» отнесен к группе прецедентов работы с отдельной задачей. Но такая операция с объектом «Задача» невозможна в объектно-ориентированной ИС.

В таблице 3.1 представлены типичные ошибки, допускаемые студентами при создании UCD с High LoD. Ошибки, подобные 1.1 -1.7, 1.10 – 1.17, 1.21, которые представлены в таблице 1.1, повторно не рассматриваются. Для ошибок 1.8 – 1.9, 1.18 – 1.20 приведены дополнительные примеры UCD или их фрагменты (номер ошибки привязан к номеру таблицы и начинается с символа «3»: это ошибки 3.8 – 3.9, 3.18 – 3.20).

Второе число в номере ошибок в таблице 3.1 продолжает нумерацию ошибок UCD, начатую в таблице 1.1.

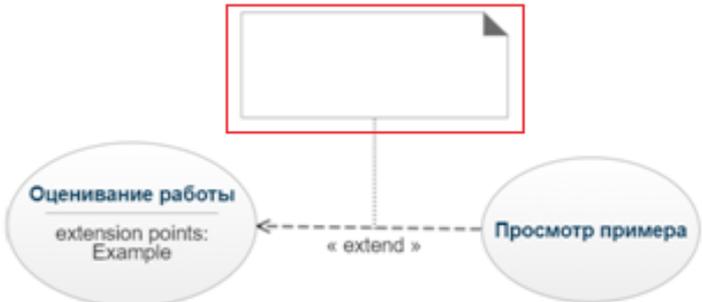
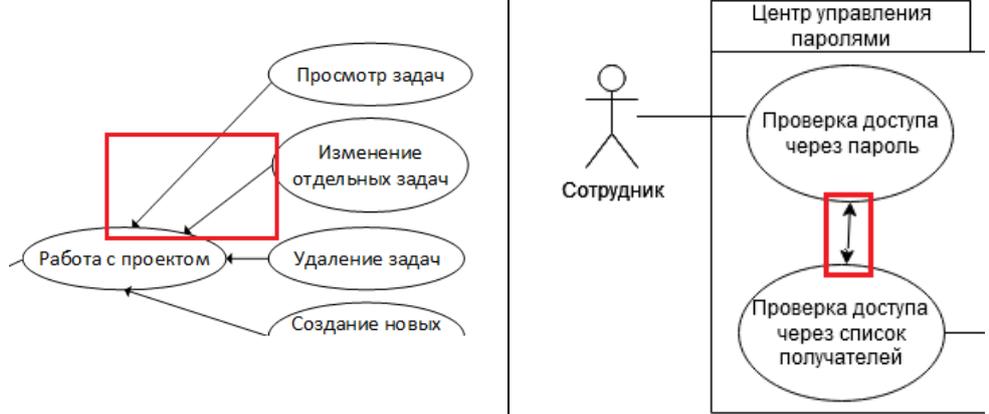
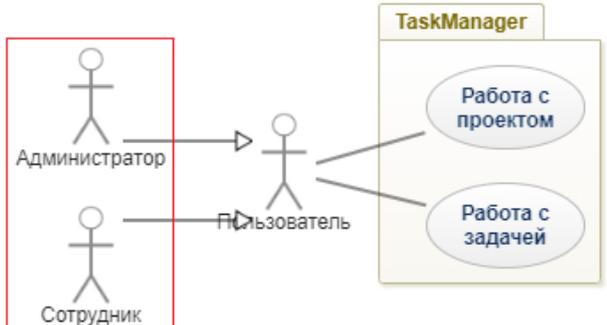
Таблица 3.1 – Ошибки в диаграммах прецедентов с высоким уровнем детализации

Номер и текст ошибки	Пример с ошибкой	
1	2	
Лексические ошибки		
3.8 Неверное имя прецедента		
Имя прецедента должно быть представлено в виде действия (глаголом в инфинитиве или именем существительным отглагольной формы), начинаться с заглавной буквы и не должно содержать символы, недопустимые в идентификаторах		
3.9 Перемешивание в названиях прецедентов глаголов и имен существительных		
Следует придерживаться во всех UC единого стиля написания имен: или глаголы, или существительные		
3.22 Неверный символ включения		
GenMyModel не позволит создать такие элементы при использовании панели «Use case diagram»		

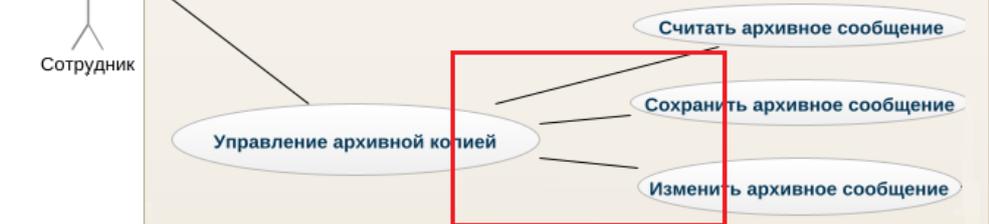
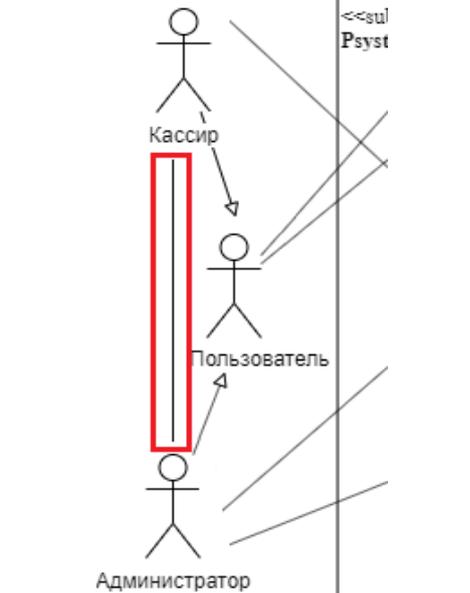
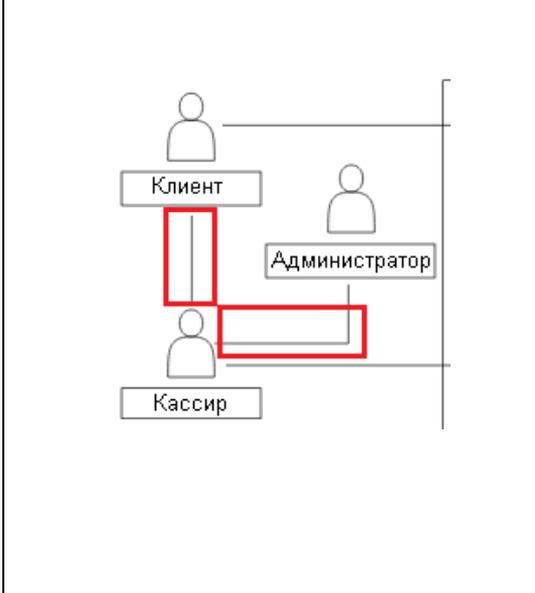
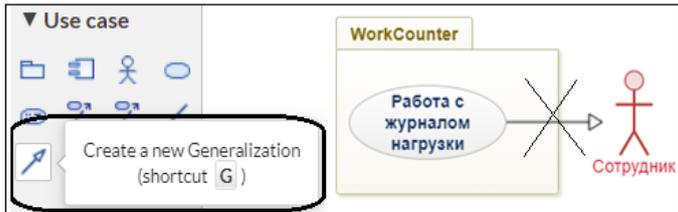
Продолжение таблицы 3.1

1	2
<p>3.23 Неверный символ обобщения</p>	
<p>3.23 Неверный символ обобщения</p>	
<p>GenMyModel не позволит создать такие элементы при использовании панели «Use case diagram» ни между акторами, ни между прецедентами</p>	
<p>3.24 Отсутствие точки расширения у прецедента с зависимостью расширения</p>	
<p>Указание точек расширения у прецедента с зависимостью расширения предусмотрено стандартом UML. Точки расширения обозначаются ключевыми словами «extention points:»</p>	
<p>3.25 Отсутствие текста в точке расширения прецедента</p>	
<p>Спецификация точки расширения прецедента в виде текста предусмотрена стандартом UML. Значение каждой точки расширения записывается после ключевых слов «extention points:»</p>	

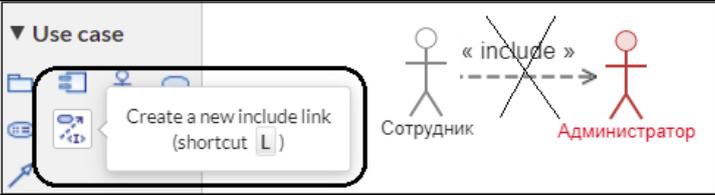
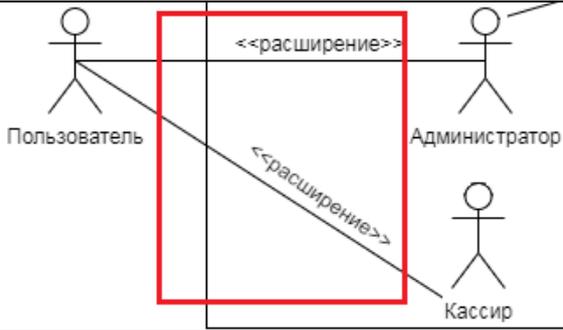
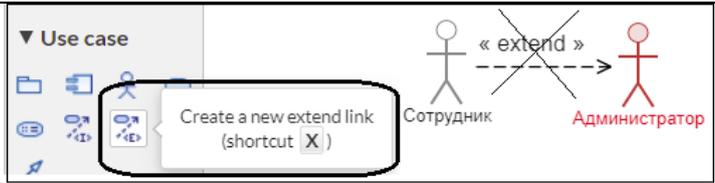
Продолжение таблицы 3.1

1	2
<p>3.26 Отсутствие текста в условии расширения прецедента</p>	
<p>Указание условия расширения у прецедента с зависимостью расширения предусмотрено стандартом UML для каждой точки расширения. Условие расширения записывается в фигурных скобках после ключевого слова «Condition»; после ключевых слов «extension point:» указывается соответствующая точка расширения</p>	
<p>3.27 Неверный символ и/или стереотип зависимости</p>	
<p>GenMyModel не позволит создать такие элементы при использовании панели «Use case diagram»</p>	
<p style="text-align: center;">Синтаксические ошибки</p>	
<p>3.18 Актор не имеет ни одного отношения ассоциации с прецедентами</p>	
<p>Обращайте внимание при объединении нескольких диаграмм на наличие прецедентов, ассоциированных с актерами-предками и актерами-потомками: в исходных UCD акцент может быть на отношениях между актерами, но в объединенной диаграмме необходимо представить участие всех акторов в прецедентах</p>	

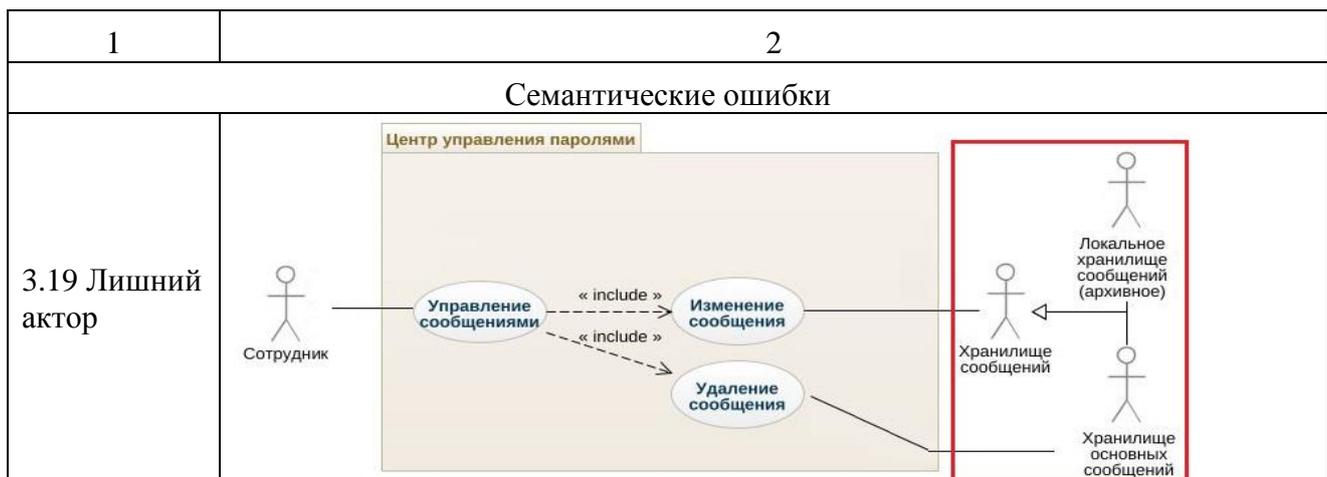
Продолжение таблицы 3.1

1	2
<p>3.28 Использование отношения ассоциации между прецедентами</p>	 
<p>Ассоциация предназначена для указания факта, что в прецеденте участвует актер. Поэтому символ ассоциации должен связывать прецедент и актора</p>	
<p>3.29 Использование отношения ассоциации между актерами</p>	 
<p>Ассоциация предназначена для указания факта, что актер участвует в прецеденте. Поэтому символ ассоциации должен связывать актора и прецедент</p>	
<p>3.30 Использование отношения обобщения не между двумя актерами или прецедентами.</p>	
<p>Отношение обобщения допустимо только между актерами или между прецедентами. GenMyModel не позволит создать такую модель, помечая образ актора красным цветом</p>	

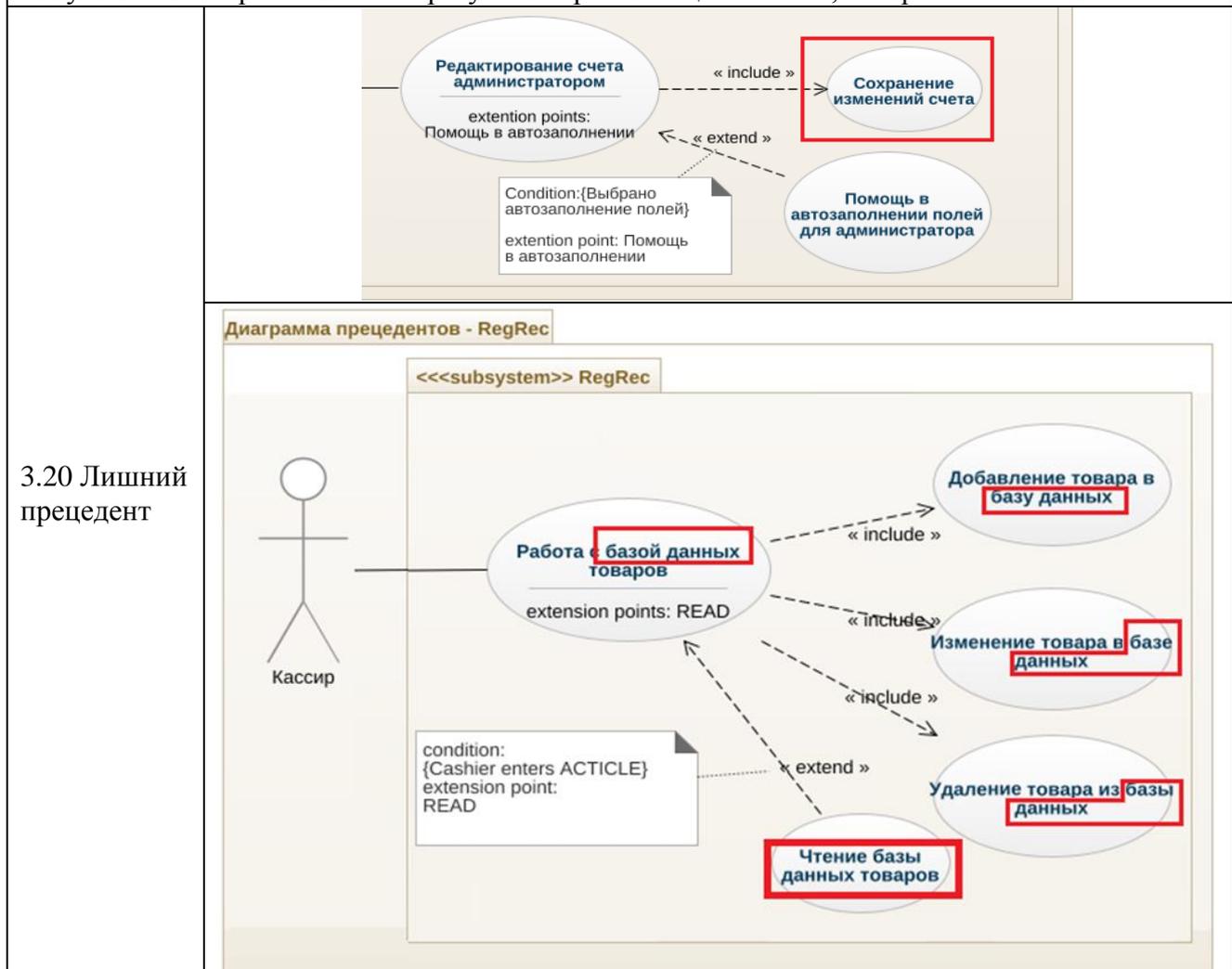
Продолжение таблицы 3.1

1	2
<p>3.31 Использование зависимости включения между акторами</p>	
<p>Зависимость включения допустима только между прецедентами. GenMyModel не позволит создать такую модель, помечая образ одного из акторов красным цветом</p>	
<p>3.32 Использование зависимости расширения между акторами</p>	
	
<p>Зависимость расширения допустима только между прецедентами. GenMyModel не позволит создать такую модель, помечая одного из акторов красным цветом</p>	
<p>3.33 Использование зависимости расширения между актором и прецедентом</p>	
	
<p>Зависимость расширения допустима только между прецедентами. GenMyModel не позволит создать такую модель, помечая образ актора красным цветом</p>	
<p>3.34 Зависимость не идентифицирована</p>	
<p>Зависимости включения и расширения должны сопровождаться именем зависимости, указанной на стрелке</p>	

Продолжение таблицы 3.1



Сущности, которые не запрашивают систему предъяснить свою функциональность и не получают некоторый полезный результат при помощи системы, актерами не являются

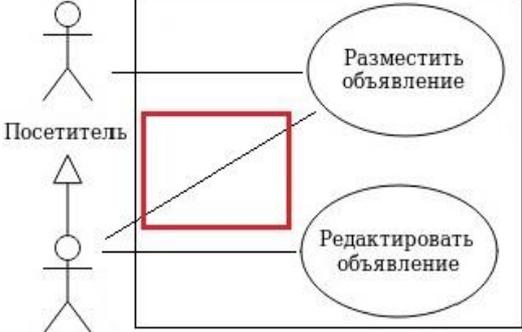
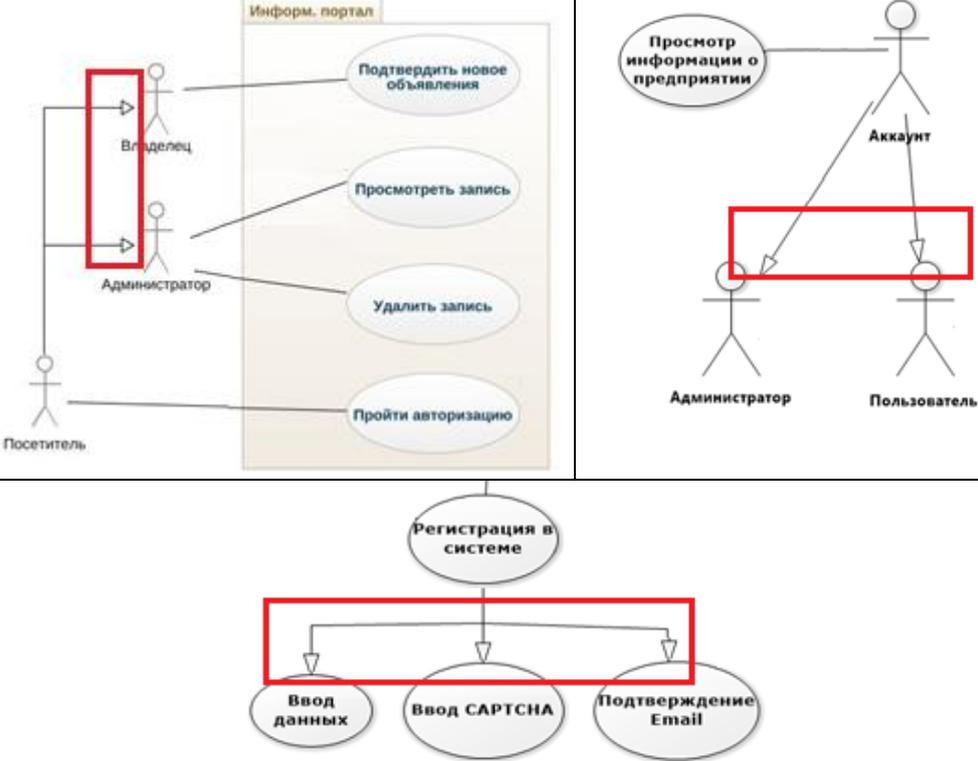


При функционировании объектно-ориентированной системы все ее объекты размещены в оперативной памяти. Изменение значений соответствующих атрибутов отдельного объекта представляет собой изменение его состояния и состояния системы в целом. Процесс «Сохранение» относится к уровню данных, но не к уровню модели. Информационные процессы, связанные с базами данных, происходят на уровне данных

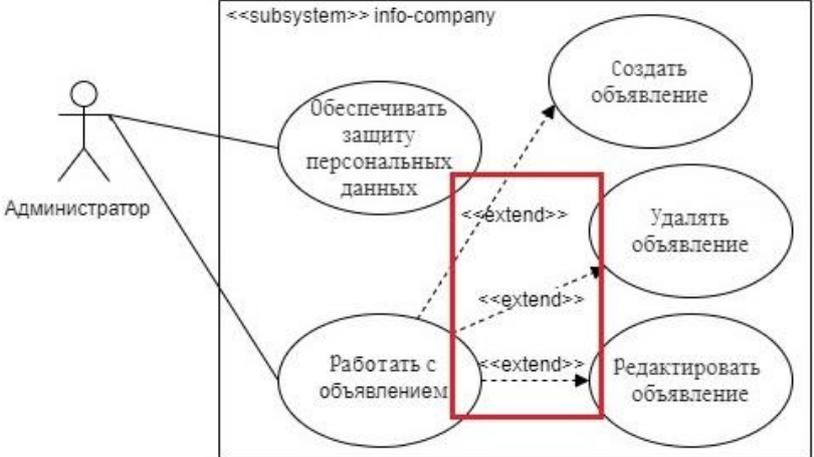
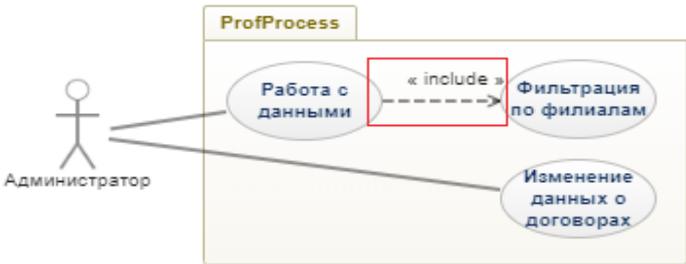
Продолжение таблицы 3.1

1	2
<p>3.35 Неиерархическая структура обобщения между акторами</p>	
<p>Отношение обобщения между акторами выражает связь «предок»-«потомок», описывая варианты участия в прецедентах для актора-предка, в которых также участвуют и акторы-потомки. Поэтому никаких иных вариантов моделей, кроме иерархической, между акторами быть не может</p>	
<p>3.36 Неиерархическая структура прецедентов</p>	
<p>Между прецедентами отношение обобщения и любой вид зависимостей должны следовать иерархической модели данных</p>	
<p>3.37 Неверная семантика ассоциации</p>	
<p>Диаграмма подготовлена с помощью инструмента, который ассоциации изображает стрелками. Стрелки есть в обе стороны (от акторов и к акторам) свидетельствуют о том, что автор модели неверно понимает семантику ассоциации. GenMyModel не позволит создать такую модель, т.к. для обозначения ассоциации используется не стрелка, а сплошная линия</p>	

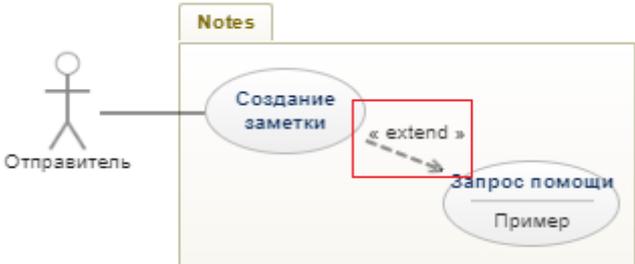
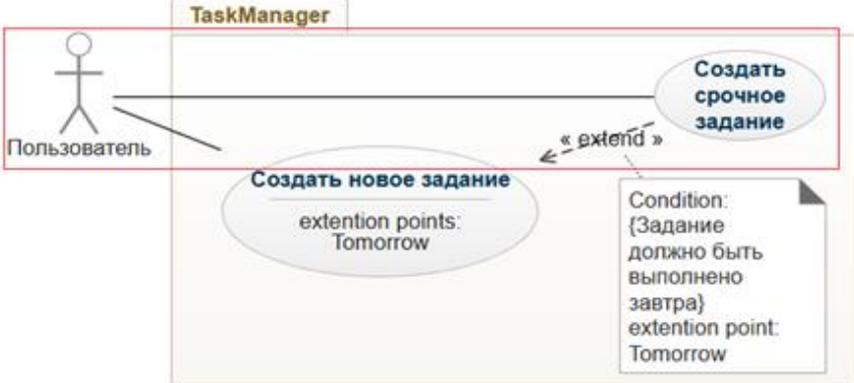
Продолжение таблицы 3.1

1	2
<p>3.38 Лишняя ассоциация</p>	 <p style="text-align: center;">Владелец предприятия</p>
<p>У актора-потомка дублируется ассоциация, уже имеющаяся у актора-предка, с одним и тем же прецедентом</p>	
<p>3.39 Обобщение направлено не в ту сторону</p>	
<p>Треугольник стрелки отношения обобщения должен быть направлен в сторону актора-предка (обобщение между акторами) или в сторону родительского прецедента (обобщение между прецедентами). Все представленные в п.39 UCD моделируют «множественное наследование» вместо одиночного, что противоречит иерархическим моделям отношений между акторами и между прецедентами</p>	
<p>3.40 Один потомок у обобщения прецедентов</p>	
<p>Отношение обобщения не должно противоречить иерархической модели (1:m)</p>	

Продолжение таблицы 3.1

1	2
<p>3.41 Зависимость включения направлена не в ту сторону</p>	
<p>Стрелка зависимости включения должна быть направлена от базового (независимого) прецедента к зависимому прецеденту</p>	
<p>3.42 Злоупотребление зависимостью включения</p>	
<p>Может свидетельствовать о том, что не все прецеденты размещены на нужном уровне иерархии прецедентов или о попытке моделировать последовательность действий</p>	
<p>3.43 Подмена зависимости включения зависимостью расширения</p>	
<p>По структуре в этой модели должны быть зависимости включения. Если же должны были быть зависимости расширения, то направление всех стрелок указано в обратную сторону, также отсутствуют точки и условия расширения</p>	
<p>3.44 Прецедент с зависимостью включения, включает всего один прецедент</p>	
<p>Зависимость включения не должна противоречить иерархической модели (1:m)</p>	

Продолжение таблицы 3.1

1	2
<p>3.45 Зависимый прецедент включения ассоциирован с актором</p>	
<p>Актор может участвовать в сценарии взаимодействия с ИС на верхнем уровне иерархии прецедентов этого сценария</p>	
<p>3.46 Зависимость расширения направлена не в ту сторону</p>	
<p>Стрелка зависимости расширения должна быть направлена от базового (расширяющего) прецедента к зависимому (расширяемому) прецеденту</p>	
<p>3.47 Подмена зависимости расширения за счет включения</p>	
<p>По структуре в этой модели должна быть зависимость расширения. В этом случае направление стрелки указано в обратную сторону, также отсутствуют точка и условие расширения. Если же должна была быть зависимость включения, то она не должна противоречить иерархической модели (1:m)</p>	
<p>3.48 Зависимый прецедент расширения ассоциирован с актором</p>	
<p>Актор может участвовать в сценарии взаимодействия с ИС на верхнем уровне иерархии прецедентов этого сценария</p>	

Продолжение таблицы 3.1

<p>1</p> <p>3.49 Родительский прецедент, имеющий зависимости включения или расширения, не имеет отношения ассоциации</p>	<p>2</p>
<p>Ассоциация предназначена для указания факта, что актер участвует в прецеденте. Поэтому символ ассоциации должен связывать родительский прецедент с актором</p>	
<p>Ошибки размещения</p>	
<p>3.50 Символ актора и символ отношения между акторами накладываются</p>	
<p>Эстетическая аномалия</p>	
<p>3.51 Символы нескольких обобщений / ассоциаций накладываются</p>	
<p>Эстетическая аномалия: может возникать как при использовании обобщения между акторами, так и у обобщения между прецедентами. Возможны наложения фрагментов символов обобщения и линий ассоциаций</p>	
<p>3.52 Изображения прецедентов расположены слишком близко друг к другу</p>	
<p>Семантическая аномалия: невозможно прочесть название стереотипа зависимости («extend» или «include») между прецедентами</p>	

4. Создание AD

В AD должны быть использованы только такие субъекты, как участники со своими дорожками, активности, начальное и конечное состояния, переходы, условные переходы, слияния, разветвители, синхронизаторы, комментарии.

На рисунке 4.1 представлен результат неверного использования инструмента GenMyModel для создания диаграммы активностей.

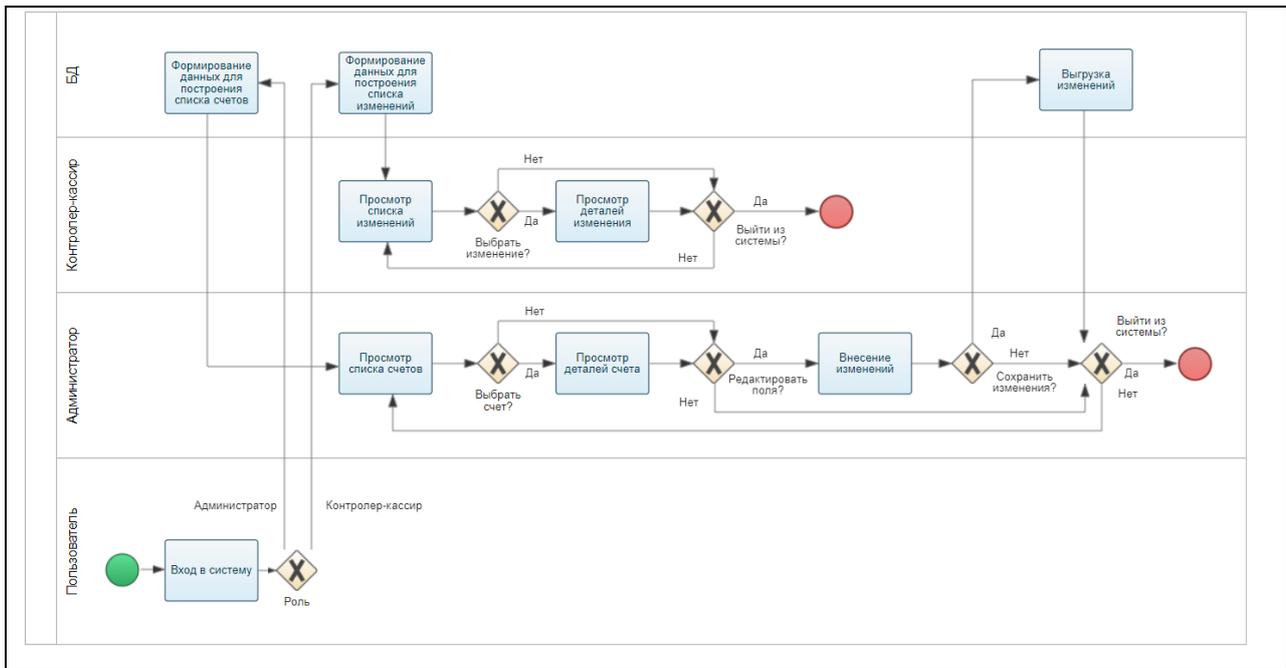


Рисунок 1 – Результат создания AD как проекта типа «BPMN Busines Process»

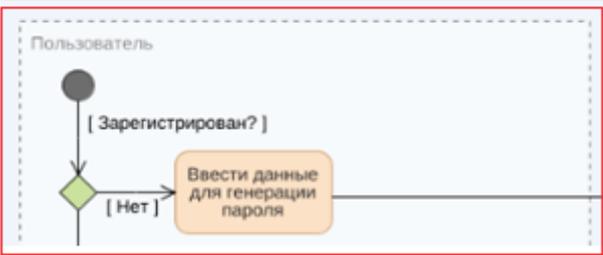
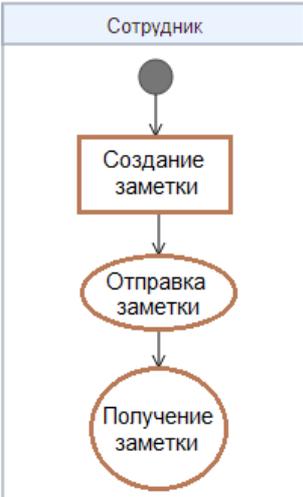
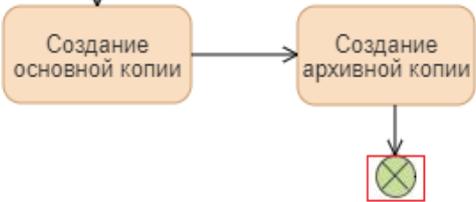
Лексическими являются ошибки, которые нарушают правила создания и использования лексем подмножества UML для создания диаграмм активностей – допустимых графических образов перечисленных выше субъектов и их именования.

Синтаксические ошибки возникают при отсутствии субъектов или их элементов, неверном установлении связей между субъектами и/или их размещении по отношению друг к другу, а также при нарушении принципа идентификации.

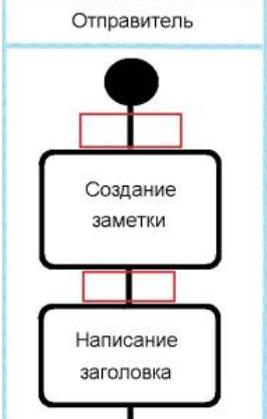
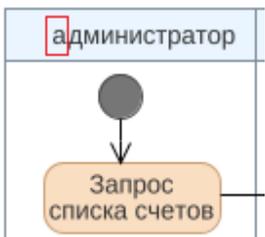
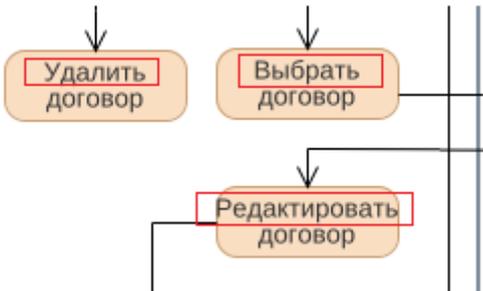
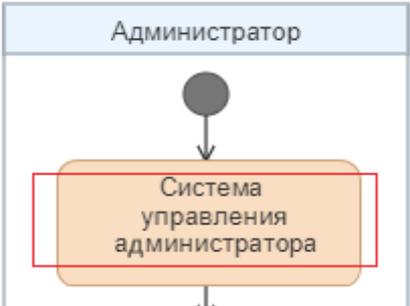
К семантическим ошибкам приводит неверное понимание моделирования бизнес-процесса (замена модели алгоритмом работы с ИС) или его независимых ветвей (с помощью синхронизаторов и разветвителей).

В таблице 4.1 представлены типичные ошибки, допускаемые студентами при создании AD.

Таблица 4.1 – Ошибки в диаграммах активностей

Ошибка	Пример с ошибкой
1	2
Лексические ошибки	
4.1 Отсутствие дорожки участника	
GenMyModel не позволит создать такую диаграмму при использовании панели «Activity diagram»	
4.2 Использование неверного элемента для дорожки участника	
GenMyModel создает дорожку участника при использовании панели «Activity diagram»	
4.3 Активности должны быть представлены соответствующим элементом AD	
Использован неверный символ активности. GenMyModel не позволит создать такие элементы при использовании панели «Activity diagram»	
4.4 Использование неверного элемента для конечного состояния деятельности	
Использован неверный символ для конечного состояния. GenMyModel позволяет создать такой элемент при использовании панели «Activity diagram», но это конечное состояние потока, которое означает завершение одного потока управления. В теме 4 следует применять элемент конечного состояния (завершение всех потоков управления внутри этой деятельности)	

Продолжение таблицы 4.1

1	2	
<p>4.5 Неверное обозначение перехода</p>		
<p>GenMyModel не позволит создать такой элемент при применении панели «Activity diagram»</p>		
<p>4.6 Неверное имя участника (объекта)</p>		
<p>Имя участника должно быть представлено именем существительным, начинаться с заглавной буквы и не содержать символы, недопустимые в идентификаторах</p>		
<p>4.7 Неверное название активности</p>		
 		
<p>Название активности должно быть представлено в виде действия (именем существительным отглагольной формы) в виде утвердительного предложения, начинаться с заглавной буквы и не содержать символы, недопустимые в идентификаторах. Указывать имя участника в названии активности не следует</p>		

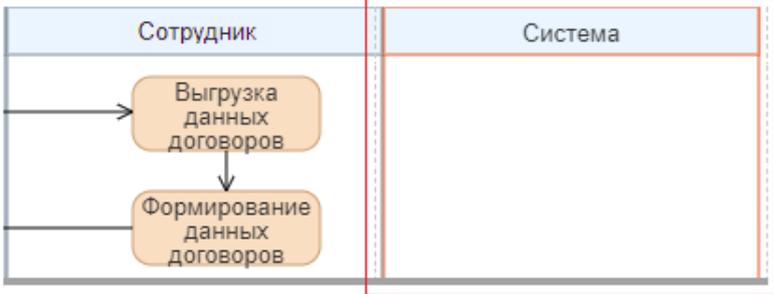
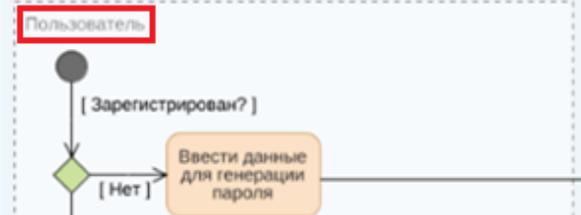
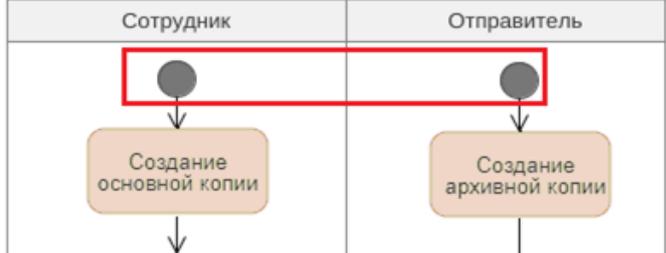
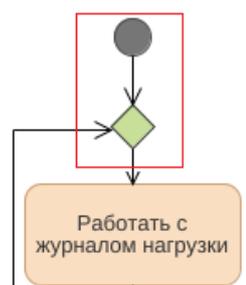
Продолжение таблицы 4.1

1	2	
<p>4.8 Размещение текста условия внутри символа условного перехода</p>		
<p>GenMyModel не позволит создать такой элемент при использовании панели «Activity diagram». Дополнительные символы в тексте альтернатив не требуются</p>		
<p>4.9 Использование недопустимого символа вместе с текстом условия в условном переходе</p>		
<p>Условный переход должен иметь текст условия и подписи альтернатив. Дополнительные элементы диаграммы для условия не требуются</p>		
<p>4.10 Отсутствие текста условия в условном переходе</p>		
<p>Условный переход должен иметь текст условия и подписи альтернатив. Отсутствие условия условного перехода не позволит принять решение в бизнес-процессе</p>		
<p>4.11 Использование неверных символов, заключающих текст условия в условном переходе, или отсутствие скобок</p>		
<p>Условный переход должен иметь текст условия, заключенный в квадратные скобки. Дополнительные символы для обрамления условия не требуются</p>		

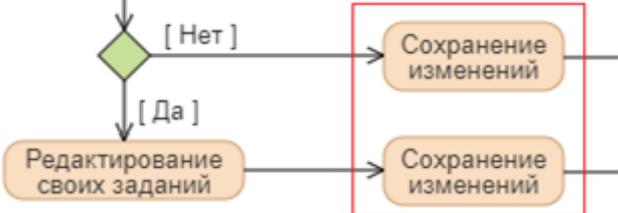
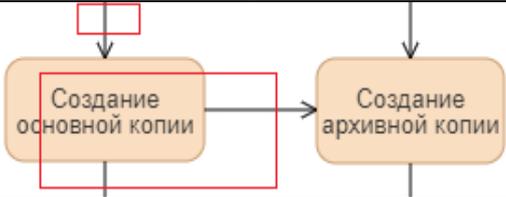
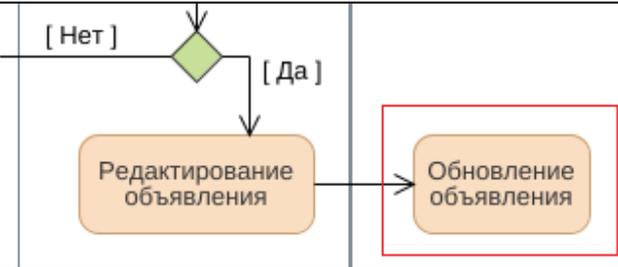
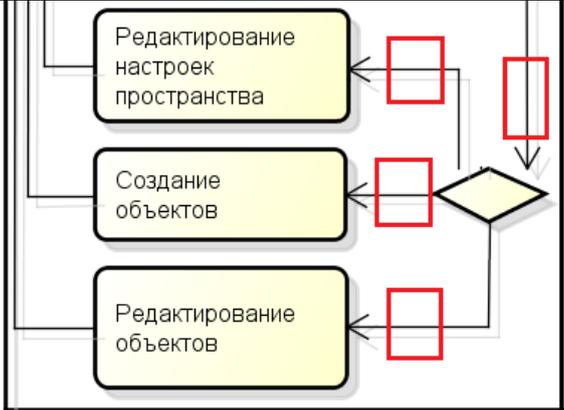
Продолжение таблицы 4.1

1	2
4.12 Неверный текст условия у условного перехода	
Отсутствие знака вопроса в конце текста условия	
4.13 Отсутствие текста альтернативы условного перехода	
Условный переход должен иметь текст условия и подписи альтернатив. Отсутствие подписи альтернативы условного перехода не позволит принять решение в бизнес-процессе	
4.14 Неверный текст альтернативы условного перехода	
Текст альтернатив условного перехода заключают в квадратные скобки. Указывать вопросительный знак в альтернативе не следует	
4.15 Переход помечен альтернативой	
Переход между активностями имеет последовательную природу и выполняется в соответствии с бизнес-процессом. Стрелку перехода может сопровождать только комментарий как в модели 4.15, б)	
4.16 Неверный разветвитель	
Разветвитель не требует текста условия, его ветви не требуют текста альтернатив. Ошибка свидетельствует о подмене условного перехода разветвителем	

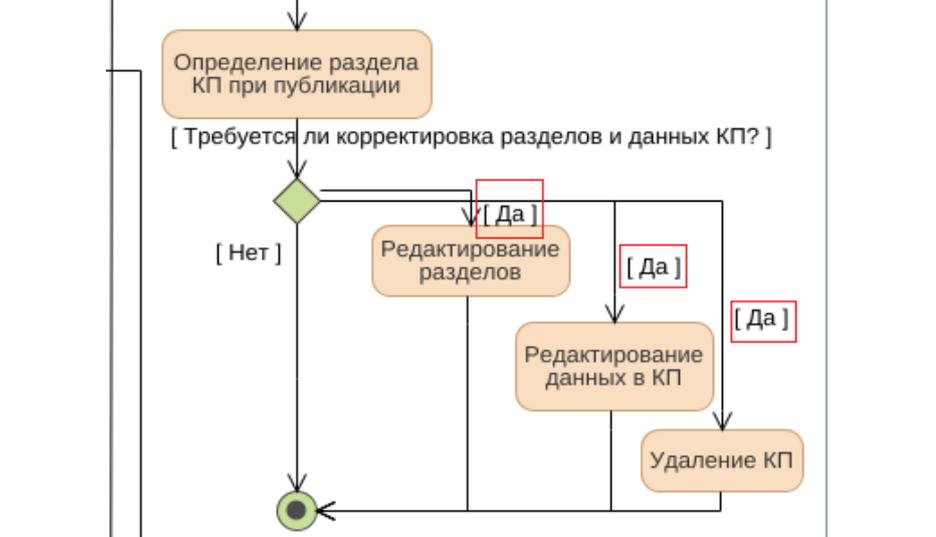
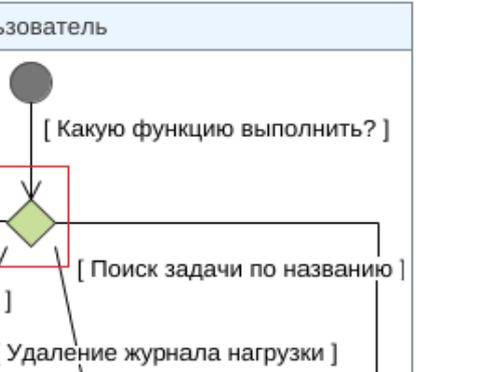
Продолжение таблицы 4.1

1	2	
Синтаксические ошибки		
4.17 Пустая дорожка участника		
Не удалена дорожка участника после удаления элементов из нее		
4.18 Размещение имени участника внутри дорожки		
Имя участника должно быть размещено над его дорожкой		
4.19 Более одного начального состояния		
Бизнес-процесс имеет ровно одну точку входа		
4.20 Отсутствие точки инициализации или конечного состояния		
Подмена начального или конечного состояния элементом «активность»		
4.21 Размещение условного перехода или слияния непосредственно после входного состояния		
После начального состояния бизнес-процесс должен перейти к действию в состоянии соответствующей активности, а затем может следовать принятие решения		

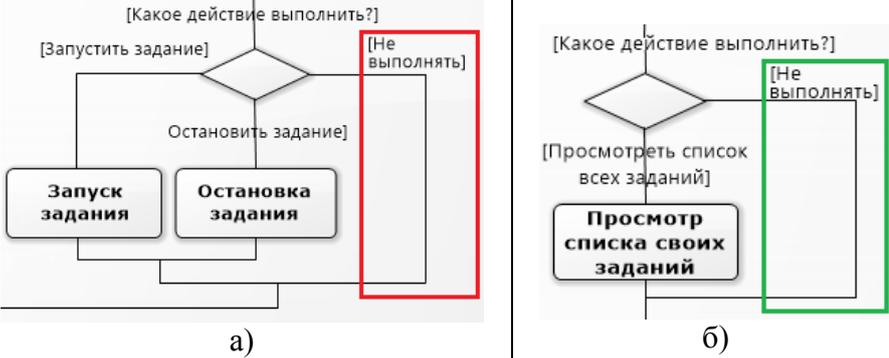
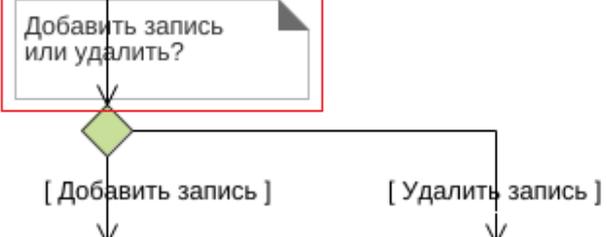
Продолжение таблицы 4.1

1	2
<p>4.22 Названия активностей должны быть уникальными</p>	
<p>Обращайте внимание на названия активностей при объединении нескольких диаграмм активностей: в исходных АД названия активностей могут быть уникальными, но в объединенной диаграмме это свойство может быть утеряно. Использование активности с названием «Сохранение ...» обсуждается в ошибке 4.40, ж).</p>	
<p>4.23 Несоответствие количества входящих и выходящих переходов у активности</p>	
<p>Активность не может иметь больше одного выходящего перехода</p>	
<p>4.24 Отсутствие выходящего перехода из активности</p>	
<p>Активность не может выступать в роли конечного состояния с входящим переходом, но без выходящего</p>	
<p>4.25 Неверный условный переход</p>	
<p>У условного перехода отсутствуют условие и альтернативы. Условный переход описывает принятие решения в бизнес-процессе, поэтому необходимо указывать тексты условия и альтернатив для представления в модели принятия решения. Отсутствие текстов альтернатив свидетельствуют о том, что автор модели заменяет их названиями активностей</p>	

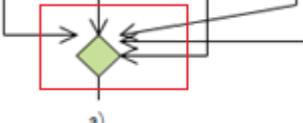
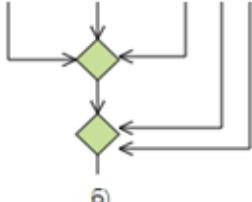
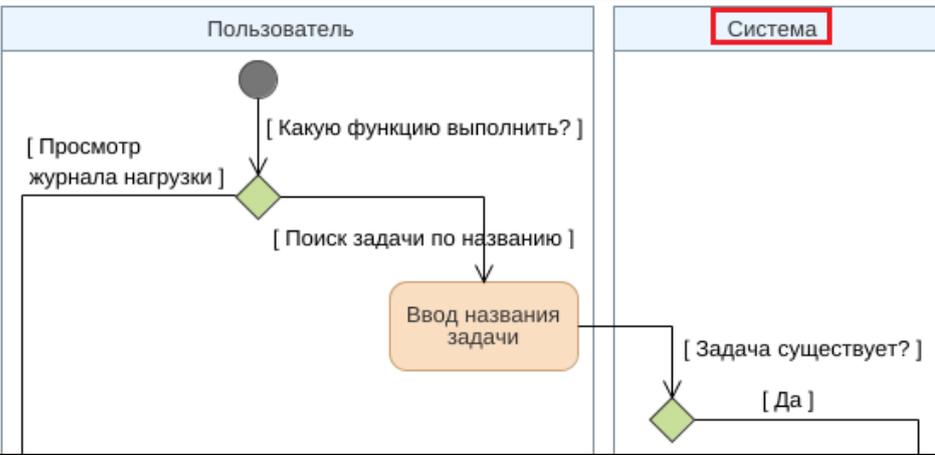
Продолжение таблицы 4.1

1	2
<p>4.26 Тексты условий условных переходов должны быть уникальными</p>	
<p>Условный переход описывает принятие решения в бизнес-процессе, поэтому условия для принятия решений не должны дублироваться. В этой АД название активности «Удаление...» соответствует контексту деятельности со множеством сущностей (списком получателей)</p>	
<p>4.27 Значения альтернатив одного условного перехода должны быть уникальными</p>	
<p>Условный переход описывает принятие решения в бизнес-процессе, поэтому значения альтернатив одного условного перехода должны соответствовать различным вариантам протекания бизнес-процесса. У различных условных переходов значения альтернатив могут совпадать, напр., как в ошибках 4.20, 4.22 и 4.24</p>	
<p>4.28 Использование больше трех альтернатив у условного перехода</p>	
<p>Условный переход имеет не более трех альтернатив. Некоторые приложения для создания UML диаграмм имеют инструмент типа «SelectCase» для ветвления на любое количество направлений. В общем случае для реализации ветвления более, чем на три направления, в АД размещают несколько подряд идущих условных переходов. Однако не следует отождествлять условный переход с блоком ветвления блок-схем (напр., ошибка 4.29)</p>	

Продолжение таблицы 4.1

1	2
4.29 Использование одной альтернативы у условного перехода	
<p>Условный переход имеет только одну альтернативу. Условный переход описывает принятие решения в бизнес-процессе, поэтому альтернатив должно быть не менее двух</p>	
4.30 Использование альтернативы условного перехода без активности	
<p>Альтернатива условного перехода без активности избыточна, ее можно в AD не показывать (ошибка 4.30, а). У условного перехода с двумя альтернативами одна из них может быть без активности (ошибка 4.30, б)</p>	
4.31 Использование нескольких условных переходов подряд	
<p>По каждой альтернативе бизнес-процесс должен перейти в какое-то состояние. Это не происходит в случае последовательного размещения нескольких условий по любой из альтернатив</p>	
4.32 Использование комментария вместо условия в условном переходе	
<p>xmi-файл модели бизнес-процессов при необходимости может быть обработан приложениями проверки правильности модели, ее интерпретации или оценки сложности. Поэтому «условие» в условном переходе – это не просто текст, который визуально должен предшествовать символу ромба, это значение соответствующего тэга в файле модели. Текст комментария указанными выше приложениями не используется, он требуется для повышения удобства чтения модели и восприятия последней пользователями</p>	

Продолжение таблицы 4.1

1	2	
4.38 Неверное использование элемента слияния	 <p style="text-align: center;">а)</p>	 <p style="text-align: center;">б)</p>
У элемента слияния может быть использовано до трех входов. Здесь 4.38, а) – это ошибка, 4.38, б)– это ее исправление		
Семантические ошибки		
4.39 Несуществующая активность		
Удаление договора участником не предусмотрено, но присутствует в бизнес-процессе. Также AD отображает противоречие бизнес-процесса: в нем удалить договор можно только после его редактирования		
4.40 Подмена понятия «бизнес-процесс» понятием «алгоритм работы с ИС»: а) использование Системы как участника		
Система не может быть участником бизнес-процесса, ИС – это инструмент достижения цели организации		
б) использование лишней активности		
Активность «Вход в систему» не является частью бизнес-процесса		

Продолжение таблицы 4.1

<p>1</p> <p>в) использование лишней активности как конечного состояния бизнес-процесса</p>	
<p>Активность «Выход из системы» не является частью бизнес-процесса</p>	
<p>г) использование лишней активности перед конечным состоянием бизнес-процесса</p>	
<p>Активность «Закрытие программы» не является частью бизнес-процесса</p>	
<p>д) использование лишнего участника и ассоциированной с ним активности в бизнес-процессе</p>	
<p>Активность «Аутентификация пользователя» в общем случае не является частью бизнес-процесса. При удалении ее из модели лишней становится участник «Пользователь» и соответствующая ему дорожка. При удалении из модели дорожки «Пользователя» бизнес-процессы остальных участников становятся независимыми друг от друга.</p>	
<p>Поэтому должны быть две модели для каждого из участников</p>	

Продолжение таблицы 4.1

1	2		
е) использование базы данных как участника			
База данных не может быть участником бизнес-процесса, это сущность уровня данных, но не уровня модели			
ж) использование лишней активности не из бизнес-процесса			
Активность «Сохранение данных» не является частью бизнес-процесса. Указанное действие ассоциируется с уровнем данных, но не с уровнем модели. На уровне объектно-ориентированной модели все операции реализуются в объектах текущей сессии выполнения приложения, размещенных в оперативной памяти			
Ошибки, связанные с использованием разветвителя и синхронизатора			
4.41 Использование разветвителя без синхронизатора			
Разветвитель и синхронизатор – это рамочные (парные) элементы AD. Раздельное их использование является неверным.			
4.42 Использование синхронизатора без разветвителя			
Синхронизатор и разветвитель – это рамочные (парные) элементы AD. Раздельное их использование является неверным. Ошибка свидетельствует о подмене элемента слияния синхронизатором			

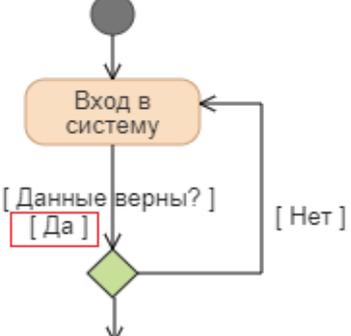
Продолжение таблицы 4.1

1	2
<p>4.43 Подмена слияния синхронизатором</p>	
<p>Даже при использовании парных разветвителя и синхронизатора такое использование последнего не является верным</p>	
<p>4.44 Альтернатива условного перехода выходит к началу зоны между парными разветвителем и синхронизатором</p>	
<p>Возможна повторная активация уже активного состояния</p>	
<p>4.45 Альтернатива условного перехода выходит за пределы зоны между парными разветвителем и синхронизатором</p>	
<p>Не все переходы между активностями, активированными разветвителем, гарантируют завершение активностей до соответствующего синхронизатора</p>	

Продолжение таблицы 4.1

1	2
<p>4.46 Бизнес-процесс после разветвителя, не имеющий активностей</p>	
<p>Назначение разветвителя состоит в активации больше чем одного состояния</p>	
<p>4.47 Альтернатива условного перехода входит внутрь зоны между парными разветвителем и синхронизатором</p>	
<p>Переход в активность, размещенную внутри зоны между парными разветвителем и синхронизатором, не допускается</p>	
<p>4.48 Не согласовано количество разветвителей и синхронизаторов</p>	
<p>Разветвитель и синхронизатор – это рамочные (парные) элементы AD</p>	
<p>4.49 Синхронизатор никогда не будет активирован</p>	
<p>Семантическая ошибка: продолжение бизнес-процесса возможно только при приобретении пакетов ВСЕХ трех видов, хотя в описании предметной области предполагается выбор пакета только одного вида</p>	

Продолжение таблицы 4.1

1	2
<p>4.50 Наложение перехода в элемент слияния и перехода от разветвителя</p>	<p style="text-align: center;">Ошибки размещения</p> 
<p>4.51 Наложение перехода в условный переход и его альтернативы</p>	
<p>4.52 Сдвинута подпись альтернативы</p>	
<p>Подпись альтернативы сдвинута вверх к условию условного перехода</p>	
<p>4.53 Неверное расположение бизнес-процессов различных участников</p>	
<p>Элементы, моделирующие бизнес-процессы различных участников, должны быть размещены параллельно друг другу («плавательные дорожки бассейна»). Такая ошибка возможна из-за лексической ошибки 4.2. Ошибка, связанная с БД, описана в 4.40, e).</p>	

5. Концептуальное моделирование

Имена понятий рекомендуется начинать с заглавной буквы, т.к. на этапе проектирования они в большинстве случаев трансформируются в имена соответствующих классов. По этой же причине имеет смысл сразу все имена (понятий и их атрибутов) записывать в венгерской нотации.

Обязательно должно быть выявлено отношение обобщения между понятиями ПрО, а не только между акторами. Отсутствие обобщения между понятиями является серьезной семантической ошибкой.

В таблице 5.1 представлены типичные ошибки, допускаемые студентами при выполнении концептуального моделирования.

Таблица 5.1 – Ошибки в концептуальном моделировании

Номер и текст ошибки	Пример с ошибкой	
1	2	
Ошибки при непосредственном выделении понятий		
5.1 Имена понятий должны быть именами существительными	Редактировать объявление	
Термин «Понятие» определяют как представление некоторой идеи или объекта в сознании человека. Поэтому для его обозначения используют имена существительные		
5.2 Отсутствие понятия, указанного в описании ПрО	Портал Пользователь Администратор Гость Предприятие Договор ...	ДатаОткрытия ДатаЗакрытия ТестовыйПакетУслуг СтандартныйПакетУслуг VIPПакетУслуг СрокОплаты ...
В описании ПрО указано: «...О каждом предприятии известно: срок оплаты (информация размещается на портале на платной основе, срок может быть не указан, если выбран тестовый пакет), название предприятия, адрес, номер телефона, факс, e-mail, имя контактного лица, адрес сайта, описание предприятия, логотип, пакет (тестовый, стандартный, VIP – определяет количество дополнительных услуг для данного предприятия), список рубрик, в которых расположено предприятие, логин пользователя, который управляет этим предприятием...». В описании ПрО есть «Список рубрик», но в списке понятий его нет		
5.3 Использование «искусственного» понятия, которого нет в описании ПрО	КлючДоступа – ключ для получения доступа к сообщению, который генерируется в ИС	
	Ключ доступа – ключ, необходимый для получения доступа к секретному сообщению, генерируется на стороне БД	
В описании ПрО указано: «...Доступ к сообщениям возможен, только если сотрудник указал свой пароль доступа к тайным сообщениям...», но нет прямой информации о ключах доступа.		

Продолжение таблицы 5.1

1	2	
5.4 Имена понятий должны быть уникальными	Ключ доступа – ключ для получения доступа к сообщению, который генерируется в ИС	
	Ключ доступа – ключ, необходимый для получения доступа к секретному сообщению, генерируется на стороне БД	
Нет возможности правильно идентифицировать каждое из понятий, т.к. они имеют одинаковые имена		
Ошибки после поиска по списку стандартных категорий понятий		
5.5 Неверная классификация понятия в категории «Физические или материальные объекты»	Категория понятий	Имена претендентов на роль понятий
	Физические или материальные объекты	Пользователь
	Физические или материальные объекты	Объявление, Информация о предприятии, Информация о владельце предприятия
Физические или материальные объекты	Список договоров, Договор, Закрытый договор, Открытый договор, Акт о возврате предмета залога, Чек, Предмет залога	
<p>К категории понятий «Физические или материальные объекты» относят те, которые реально существуют, могут быть наблюдаемыми, измеряемыми и т. п., напр., физический объект «Земля» (и ее модель – глобус) или материальный объект «Самолёт» (и его модель в аэродинамической трубе).</p> <p>а) «Пользователь» правильнее отнести к категории «Роли людей»,</p> <p>б) «Объявление» относится к категории «Содержимое контейнеров» (но контейнер не идентифицирован), а «Информация о предприятии» и «Информация о владельце предприятия» – это атрибуты «Объявления»,</p> <p>в) «Список договоров» относится к категории «Контейнеры других объектов», а «Договор» – к категории «Содержимое контейнеров»,</p> <p>г) «Акт о возврате предмета залога» и «Чек» можно отнести к категории "Записи финансовой, трудовой, юридической и др. деятельности"</p>		
5.6 Неверная классификация понятия в категории «Места»	Категория понятий	Имена претендентов на роль понятий
	Места	IS-FiscalRegistrator
Категория понятий «Места» – это локации, где размещаются экземпляры понятий "Физический или материальный объект". Тогда согласно приведенной классификации человек-оператор ИС «IS-FiscalRegistrator» с ролью Пользователь размещен внутри этой ИС, что противоречит концепции любой ИС		
5.7 Неверная классификация понятия в категории «События»	Категория понятий	Имена претендентов на роль понятий
	События	Сдача
Понятия необходимо отнести к категории «События», если ИС моделирует какие-то процессы и события (напр., модель операционной системы). «Сдача» не является понятием, это – атрибут		

Продолжение таблицы 5.1

1	2	
5.8 Неверная классификация понятия в категориях «Транзакции» и «Элемент транзакции»	Категория понятий	Имена претендентов на роль понятий
	Транзакции	Отправка сообщения
	Элемент транзакции	Секретное сообщение
<p>В описании ПрО указано: «...Центр управления паролями. ИС предназначена для обмена секретными сообщениями и хранения паролей...». Наиболее общее толкование термина «Транзакция» подчеркивает, что такая операция должна иметь смысл и может быть совершена только полностью. Однако доставка сообщения при его отправке не обязательно будет обеспечена. В описании этой ПрО нет детализации процесса отправки. Поэтому в модели необходимо сосредоточиться на использовании секретных сообщений пользователями, а не на процессе отправки</p>		
5.9 Неверная классификация понятия в категории «Контейнеры других объектов»	Категория понятий	Имена претендентов на роль понятий
	Контейнеры других объектов	Каталог объявлений, Каталог предприятий, Каталог владельцев предприятий
<p>Выделены претенденты на роль понятий-контейнеров, но в классификации отсутствуют понятия, являющиеся содержимым этих контейнеров</p>		
5.10 Неверная классификация понятия в категории «Содержимое контейнеров»	Категория понятий	Имена претендентов на роль понятий
	Контейнеры других объектов	Интернет-портал
	Содержимое контейнеров	Список рубрик, Пакет, Тестовый пакет, Платный пакет, Рубрика
<p>Рассматривается ИС «Интернет-портал с информацией о предприятиях», описание ПрО которой приведено в ошибке 5.2. Интернет-портал не является контейнером для тех понятий, имена которые перечислены в столбце претендентов на роль понятий. Кроме этого, здесь смешаны понятия, их атрибуты и значения этих атрибутов:</p> <p>а) «Список рубрик» – это контейнер других объектов, б) «Рубрика» – это содержимое контейнера «Список рубрик», в) «Пакет» – это атрибут примитивного перечислимого типа, а не понятие, г) «Тестовый пакет» и «Платный пакет» – это значения атрибута «Пакет»</p>		
5.11 Неверная классификация понятия в категории «Каталоги»	Категория понятий	Имена претендентов на роль понятий
	Каталоги	Список Товаров
<p>Понятия категории «Каталоги» имеют статическую природу и предназначены для получения каких-то данных, которые в них зафиксированы. Понятие «СписокТоваров» подразумевает добавление элементов в него (товары) в процессе продажи, поэтому его правильнее отнести к категории «Контейнеры других объектов»</p>		

Продолжение таблицы 5.1

1	2	
5.12 Неверная классификация понятия в категории «Записи финансовой, трудовой и др. деятельности»	Категория понятий Записи финансовой, трудовой и др. деятельности	Имена претендентов на роль понятий Договор, СрокОплаты, ЗакрытыйДоговор
«СрокОплаты» не является понятием, это – атрибут примитивного типа		
5.13 Неверная многомерная классификация понятий	Категория понятий	Имена претендентов на роль понятий
	а) транзакции	Отправка объявления, Редактирование объявления, Оценивание объявления, Выбор пакета
	б) элемент транзакции	Пакет, Тестовый пакет, VIP пакет, Стандартный пакет, Предприятие, Список предприятий
	в) контейнеры других объектов	Интернет-портал
	г) содержимое контейнеров	Пакет, Тестовый пакет, VIP пакет, Стандартный пакет, Предприятие, Список предприятий
Одни и те же понятия отнесены одновременно к различным категориям понятий: в а)-б) они интерпретируются как динамические (элементы процесса), а в в)-г) – как статические		
5.14 Понятие с несколькими атрибутами не является контейнером	Категория понятий	Имена претендентов на роль понятий
	а) контейнеры других объектов	КарточкаПредприятия
	б) содержимое контейнеров	ДоговорНаРазмещениеНаПортале, ПакетУслуг, ЛоготипПредприятия
	в) контейнеры других объектов	ФискальныйЧек, АртикулТовара, Товар
	г) содержимое контейнеров	стандартныйЗаголовок, стандартноеОкончание, имяКассира, имяПокупателя, названиеТовара, ценаЗаЕдиницу, ставкаНДС, датаФиксации, кол-воПриобренногоТовара, общаяСтоимость, НДС
<p>а)-б): Рассматривается ИС «Интернет-портал с информацией о предприятиях», описание ПрО которой приведено в ошибке 5.2. Перечисленные имена претендентов на роль понятий из категории «Содержимое контейнеров» являются атрибутами понятия «КарточкаПредприятия»: «ПакетУслуг» имеет примитивный тип данных (перечисление), «ДоговорНаРазмещениеНаПортале» и «ЛоготипПредприятия» могут моделироваться как самостоятельные понятия, но они имеют различную природу. Поэтому «КарточкаПредприятия» не является контейнером.</p> <p>в)-г): В описании ПрО указано: «...О каждом чеке известно: список товаров, имя покупателя, имя кассира, дата, общая стоимость, НДС. О каждом товаре в чеке известно: название товара, цена за единицу, количество приобретенного товара, общая стоимость, НДС. ...». Перечисленные имена претендентов на роль понятий являются атрибутами примитивных типов данных (строки, дата/время, числовое значение), но не самостоятельными понятиями одной природы</p>		

Продолжение таблицы 5.1

1	2	
5.15 Значения атрибутов перечислимого типа данных не являются самостоятельными понятиями	Категория понятий	Имена претендентов на роль понятий
	Содержимое контейнеров	ТестовыйПакетУслуг, СтандартныйПакетУслуг, VIPПакетУслуг
«ТестовыйПакетУслуг», «СтандартныйПакетУслуг», «VIPПакетУслуг» – это значения атрибута (напр., «ПакетУслуг») с типом данных перечисление		
Ошибки при выявлении атрибутов понятий		
5.16 Отсутствие необходимых атрибутов	Имена понятий	Имена атрибутов
	Фискальный чек	Денег выдано, денег удержано как проценты, денег взято за пеню
В списке имен атрибутов отсутствуют дата, время, ФИО кассира, которые являются обязательными для чека		
5.17 Использование искусственных атрибутов	Имена понятий	Имена атрибутов
	Сотрудник	id, ФИО
	Пользователь	id, ФИО, логин, пароль
ОтправленнаяЗаметка	идентификатор, заголовок, тело, списокПолучателей	
Наличие «id» или «идентификатор» среди атрибутов понятий указывает на то, что выполняется анализ не на слое модели, а на слое данных. Это является серьезной семантической ошибкой		
5.18 Неверное выявление понятий и их атрибутов	Имена понятий	Имена атрибутов
	а) стандартный пакет	Описание Тарифа, Цена
	б) тестовый пакет	Описание Тарифа, Цена
	в) пакет	Типы Пакетов, Выбранный Пакет
г) VIP пакет	Описание Тарифа, Цена	
Из всех понятий а)-г) только «Пакет» действительно является понятием. У него могут быть атрибуты «Тип пакета» (с типом данных перечисление, имеющим значения «Тестовый пакет», «Стандартный пакет», «VIP пакет») и «Выбранный Пакет» (с типом данных элемента указанного перечисления). Но у понятия «Пакет» отсутствует атрибут «Описание пакета», у которого могут быть атрибуты «Описание Тарифа» и «Цена»		
5.19 Неверное выявление атрибутов у понятий, связанных отношением обобщения	Имена понятий	Имена атрибутов
	а) ПрочитаннаяЗаметка	идентификатор, заголовок, тело, списокПолучателей, датаПолучения, датаОтметки
	б) ПолученнаяЗаметка	идентификатор, заголовок, тело, списокПолучателей, датаПолучения
	в) ОтправленнаяЗаметка	идентификатор, заголовок, тело, списокПолучателей
г) ИсходнаяЗаметка	заголовок, тело, списокПолучателей	

Продолжение таблицы 5.1

1	2	
	<p>– Атрибут «идентификатор» не должен присутствовать в списке имен атрибутов понятий а)-в) согласно комментарию ошибки 5.17.</p> <p>– Понятия-потомки имеют одинаковые атрибуты. Понятие «ИсходнаяЗаметка» является супертипом по отношению к понятиям а)-в), потому дублирование у них атрибутов «заголовков», «тело» и «списокПолучателей» является серьезной семантической ошибкой – все подтипы унаследуют атрибуты как минимум предыдущего уровня иерархии обобщения.</p> <p>– Понятие «ОтправленнаяЗаметка» является подтипом понятия «ИсходнаяЗаметка», поэтому у в) должен быть характерный для этого понятия атрибут, напр., «датаОтправки», который могут наследовать подтипы понятия в).</p> <p>– Понятие «ПолученнаяЗаметка» является подтипом понятия «ОтправленнаяЗаметка», поэтому у б) должен быть характерный для этого понятия атрибут, напр., «датаПолучения», который могут наследовать подтипы понятия б).</p> <p>– Понятие «ПрочитаннаяЗаметка» является подтипом понятия «ПолученнаяЗаметка», поэтому у а) должен быть характерный для этого понятия атрибут, напр., «датаОтметки», который могут наследовать подтипы понятия а).</p> <p>– Тогда благодаря обобщению супертип г) имеет три атрибута, подтип в) – четыре, подтип б) – пять, подтип а) – шесть атрибутов. Но в списке «Имена атрибутов» следует указывать только уникальные для соответствующего понятия</p>	
<p>5.20 Неверное выявление атрибутов у понятий, связанных отношением агрегации</p>	Имена понятий	Имена атрибутов
	а) Список договоров	Договор
	б) Список непрочитанных замечток	Количество
в) Список предприятий	Карточка предприятия, количество предприятий	
<p>У понятия-контейнера должно быть хотя бы два атрибута: «множество <элементов>» (для хранения содержимого контейнера) и «количество <элементов>» (для хранения актуального количественного знания о содержимом контейнера). Понятия а) и в) могут хранить только один экземпляр вместо множества элементов, а понятие б) вообще не имеет атрибута для хранения содержимого контейнера. Также у понятия а) отсутствует атрибут, хранящий значение количества элементов в контейнере</p>		
Ошибки при выделении ассоциаций		
<p>5.21 Неверное выделение ассоциации</p>	<p>а) Закрытый договор является частью Договора (Закрытый договор – Договор)</p> <p>б) Отправитель редактирует заметку (Отправитель – Заметка)</p> <p>в) Отправитель редактирует заголовок заметки (Отправитель – Заголовок заметки)</p> <p>г) Отправитель редактирует тело заметки (Отправитель – Тело заметки)</p> <p>д) Отправитель редактирует список получателей (Отправитель – Список получателей)</p>	
<p>а): Между понятиями «Закрытый договор» и «Договор» не может быть отношения «часть-целое», т.к. документ типа «Договор» может перейти в состояние «Закрытый договор».</p> <p>б): Ассоциация выделена верно – между понятиями</p> <p>в)-д): Ассоциации выделены неверно – между понятием «Отправитель» и атрибутами «Заголовков заметки», «Тело заметки» и «Список получателей» понятия «Заметка»</p>		

Продолжение таблицы 5.1

1	2	
Ошибки после поиска по списку стандартных категорий ассоциаций		
5.22 Отсутствие классификации ассоциаций	Отправитель создает заметку (Отправитель – Заметка) Отправитель редактирует заметку (Отправитель – Заметка) Отправитель редактирует заголовок заметки (Отправитель – Заголовок заметки) Отправитель редактирует тело заметки (Отправитель – Тело заметки) Отправитель редактирует список получателей (Отправитель – Список получателей) Отправитель отправляет заметку Получателю (Отправитель – Заметка) Получатель прочитывает список заметок (Получатель – Список заметок) Получатель читает заметку (Получатель – Заметка) Получатель ставит отметку о прочтении заметки (Получатель – Заметка) Получатель редактирует список заметок (Получатель – Список заметок) Получатель архивирует заметку (Получатель – Заметка) Получатель удаляет заметку из архива (Получатель – Архив)	
Приведен только список выделенных ассоциаций, но они не классифицированы по списку стандартных категорий ассоциаций		
5.23 Неверная классификация ассоциации в категории «А является логической частью В»	Категория ассоциаций	Ассоциация
	А является логической частью В	Закрытый договор – Договор
Ошибка аналогична рассмотренной в ошибке 5.21		
Ошибки при выделении ролей ассоциаций		
5.24 Неверное выделение ролей ассоциации	«Заметка» выступает в роли «Архивированная заметка» в ассоциации «Получатель архивирует заметку (Получатель – Заметка)» «Заметка» выступает в роли «Полученная заметка» в ассоциации «Получатель читает заметку (Получатель – Заметка)»	
«Архивированная заметка» должна иметь атрибуты, характерные для этого понятия и отличающие ее от понятия «Заметка» (напр., «датаАрхивации»). «Полученная заметка» должна иметь атрибуты, характерные для этого понятия и отличающие ее от понятия «Заметка» (напр., «датаПолучения»). Поэтому «Заметка», «Архивированная заметка» и «Полученная заметка» – это не одно и то же понятие с различными ролями, это – различные понятия		
Ошибки при определении кратности ролей ассоциаций		
5.25 Отсутствие определения кратности ролей ассоциаций	Получатель, Отправитель, Архивированная заметка, Неотправленная заметка, Полученная заметка являются ролями ассоциаций	
Имена ролей ассоциаций указаны. Не указаны имена ассоциаций и их кратности. Воспользоваться этой информацией при дальнейшем моделировании невозможно		

Продолжение таблицы 5.1

1	2	
Ошибки при выявлении ограничений		
5.26 Неверные данные в ограничениях, связанных с понятиями	Имена понятий	Имена атрибутов
	Заметка	ЗаголовокЗаметки, ТелоЗаметки, ФИОПолучателя, ДатаЗаметки, ДатаОдобренияЗаметки, ОтметкаПрочтения
	<p>В ПрО существуют ограничения, связанные с понятием «Заметка»:</p> <p>а) ФИОСотрудника ≠ "",</p> <p>б) ФИОПолучателя ≠ "",</p> <p>в) ЗаголовокЗаметки ≠ NULL,</p> <p>г) ТелоЗаметки ≠ NULL,</p> <p>д) ДатаОдобренияЗаметки ≠ NULL,</p> <p>е) ОтметкаОПрочтении < ДатаОдобренияЗаметки,</p> <p>ж) ФИОСотрудника ≠ ФИОПолучателя</p>	
<p>– Среди атрибутов понятия «Заметка» нет «ФИОСотрудника», который используется в ограничениях а) и ж).</p> <p>– В ограничении е) используется «ОтметкаОПрочтении», но атрибут понятия «Заметка» имеет другое имя – «ОтметкаПрочтения».</p> <p>Все имена, использованные в ограничениях для понятия, должны быть из списка атрибутов этого понятия.</p> <p>– Также допущены ошибки: все имена атрибутов должны начинаться с прописной буквы</p>		
5.27 Неверная семантика ограничений, связанных с понятиями	Граница баллов фактора спама не может быть больше 100 и меньше 1	
Использованная в ограничении логическая операция приводит к тому, что ограничение всегда ложно		
5.28 Не выявлены ограничения, связанные с понятиями	Имена понятий	Имена атрибутов
	ИсходнаяЗаметка	заголовок, тело, списокПолучателей
	ОтправленнаяЗаметка	датаОтправки («ОтправленнаяЗаметка» – это понятие-потомок «ИсходнаяЗаметка»)
	ПолученнаяЗаметка	датаПолучения («ПолученнаяЗаметка» – это понятие-потомок «ОтправленнаяЗаметка»)
	ПрочитаннаяЗаметка	датаОтметки («ПрочитаннаяЗаметка» – это понятие-потомок «ПолученнаяЗаметка»)
<p>В ПрО существуют ограничения:</p> <p>а) заголовок ≠ "",</p> <p>б) тело ≠ "",</p> <p>в) списокПолучателей ≠ NULL,</p> <p>г) датаОтправки ≠ NULL,</p> <p>д) датаОтметки ≠ NULL</p>		
Отсутствуют ограничения, связанные с понятием «ПолученнаяЗаметка»		

Продолжение таблицы 5.1

1	2
5.29 Не выявлены ограничения, связанные с ассоциациями	Пример из ошибки 5.28
<p>Отсутствуют ограничения, связанные с обобщениями:</p> <p>а) датаОтправки <= датаПолучения (у обобщения между «ОтправленнаяЗаметка» и «ПолученнаяЗаметка»)</p> <p>б) датаПолучения <= датаОтметки (у обобщения между «ПолученнаяЗаметка» и «ПрочитаннаяЗаметка»)</p>	
Ошибки при выявлении обобщения	
5.30 Неверные понятия выделены в отношении «супертип» – «подтип»	<p>Тип отношений «супертип» – «подтип» существует между понятиями:</p> <p>Заметка:</p> <p>а) ОдобреннаяЗаметка,</p> <p>б) АрхивЗаметок</p>
<p>Понятия «Заметка» и «ОдобреннаяЗаметка» могут быть в отношении обобщения. Они представляют собой одиночные сущности, причем понятие «ОдобреннаяЗаметка» отличается от «Заметки» дополнительным атрибутом, ассоциированным со временем ее одобрения. Понятие «АрхивЗаметок» – это понятие-контейнер, поэтому между ним и одиночной сущностью не может быть обобщения, возможна только агрегация</p>	
Ошибки при выполнении многомерной множественной классификации	
5.31 Несуществующая множественная классификация	<p>1.7 Выделение отношения «супертип» - «подтип» Этот тип отношений существует между понятиями: – Договор: а) Закрытый договор.</p> <p>1.8 Выполнение многомерной множественной классификации Определим свойства разбиения супертима Договор. Разбиение является полным, т.к. Договор по истечению срока займа обязательно преобразуется в Закрытый договор. В качестве свойств разбиения укажем {complete}.</p>
<p>Многомерная множественная классификация невозможна при кратности обобщения «1:1», которая выявлена в этой ПрО между понятиями «Договор» и «Закрытый договор»</p>	
Ошибки при выявлении агрегации	
5.32 Неверное понимание отношения «часть» – «целое»	<p>1.9 Выделение отношения агрегации К коллективной агрегации отнесем понятия, которые могут существовать независимо друг от друга: а) Договор и список договоров, б) Предмет залога и договор.</p> <p>К композитной агрегации отнесем понятия, которые могут зависеть друг от друга или одно зависит от другого: – Чек и акт возврата, потому что чек формируется на основе акта возврата, – Акт возврата и договор, потому что первый опирается на информацию второго.</p>
<p>В случае коллективной агрегации отношение «часть» – «целое» между указанными понятиями присутствует: с кратностью «1:n» для а), с кратностью «1:1» для б). В случае композитной агрегации отношение «часть» – «целое» между указанными понятиями отсутствует, т.к. «Чек» не является частью «Акта возврата», а «Акт возврата» – частью «Договора». Между этими тремя понятиями иные ассоциации</p>	

6. Создание диаграмм понятий

Диаграммы понятий называют также диаграммами классов этапа анализа или CD с Low LoD [8, 9]. В обоих случаях такие названия подчеркивают, что степень детализации будет отличаться от соответствующей диаграммы классов этапа проектирования. В этом пособии используется термин «Диаграмма понятий» [13].

В диаграммах понятий для моделирования есть несколько синтаксических функций детализации понятий. Это такие субъекты, как атрибуты, имена ассоциаций, роли с их именами и кратностью, ограничения и т.д. Но в таких диаграммах с низким уровнем детализации отсутствуют операции, атрибуты представляются без каких-либо типов данных и области видимости, указание ролей ассоциаций не является обязательным.

Для разработки диаграмм понятий с помощью genMyModel используется его инструмент для создания диаграмм классов. Поэтому genMyModel добавляет после имен всех атрибутов имя типа данных по умолчанию (undefined). Это имя должно остаться в диаграммах понятий и не оценивается как верное или ошибочное. Заполнение раздела операций у понятий с помощью genMyModel в этой модели не требуется: диаграмма понятий – это структурная модель этапа анализа, а не поведенческая. Также нет необходимости указывать область видимости атрибутов в этой модели: genMyModel устанавливает для всех атрибутов значение «открытая часть класса» (символ ‘+’) по умолчанию. Этот символ должен остаться в диаграммах понятий и не оценивается как верный или ошибочный.

Все элементы концептуальной модели, выделенные и описанные в теме 5, должны быть представлены на диаграмме понятий. Поэтому необходимо соблюдать выполнение тех же требований к именам и набору отношений, которые были сформулированы для концептуального моделирования.

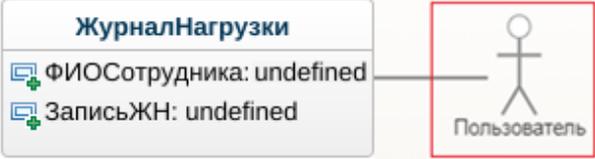
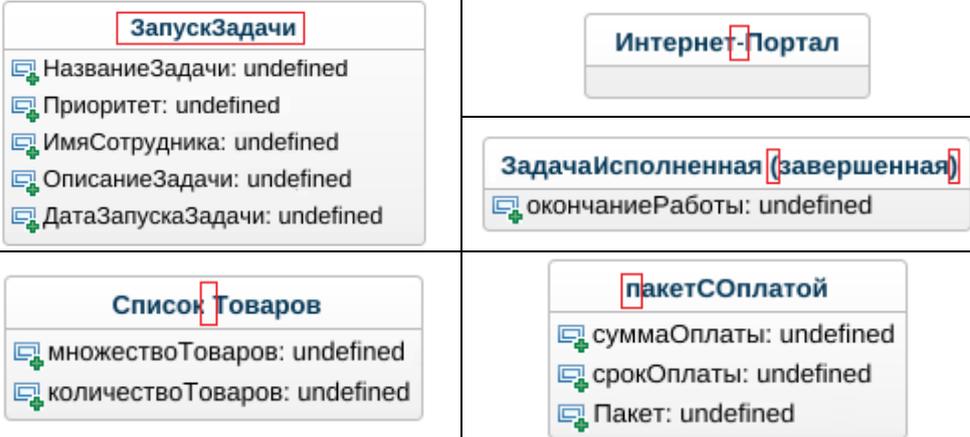
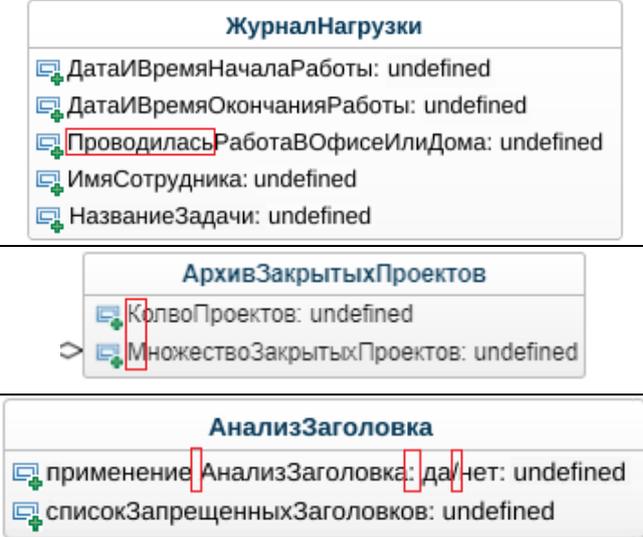
Лексическими являются ошибки, которые нарушают правила создания и использования лексем подмножества UML для создания диаграмм понятий – допустимых графических образов понятий и перечисленных выше субъектов и их именования.

Синтаксические ошибки возникают при отсутствии понятий/субъектов, неверном установлении связей между понятиями и/или их размещении по отношению друг к другу, а также при нарушении принципа идентификации.

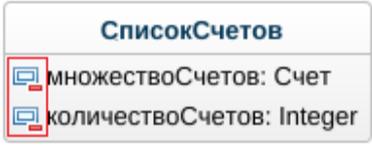
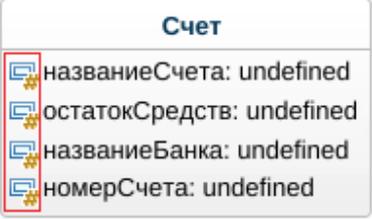
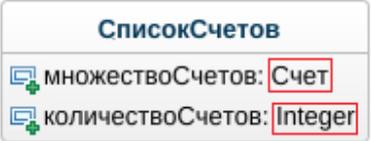
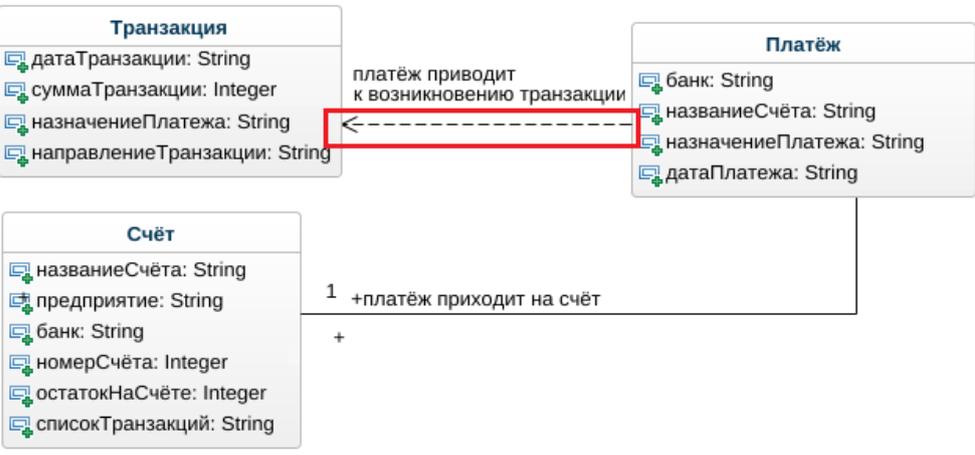
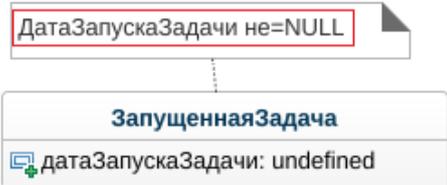
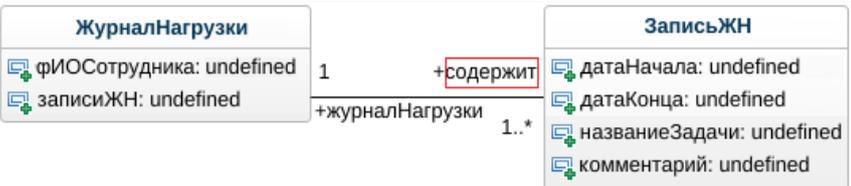
К семантическим ошибкам приводит неверное понимание моделирования отношений между понятиями, в т.ч. обобщения и агрегации.

В таблице 6.1 представлены типичные ошибки, допускаемые студентами при создании CD с Low LoD. Фрагменты диаграмм, которые представлены в этой таблице, могут содержать более одной ошибки в соответствующей строке; рассматриваемая ошибка выделена красным контуром.

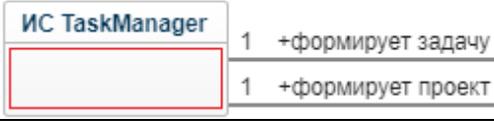
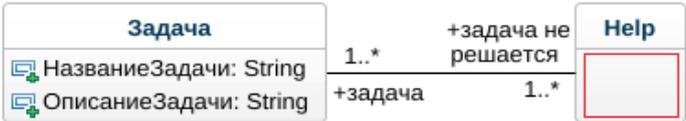
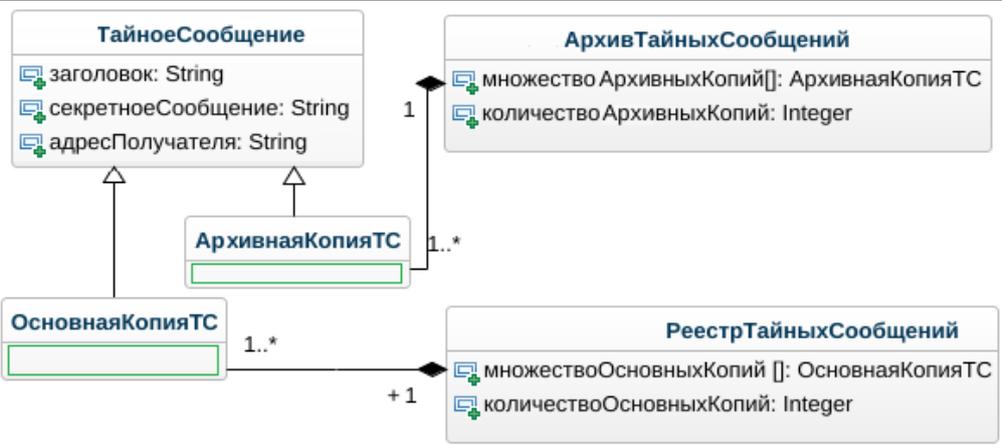
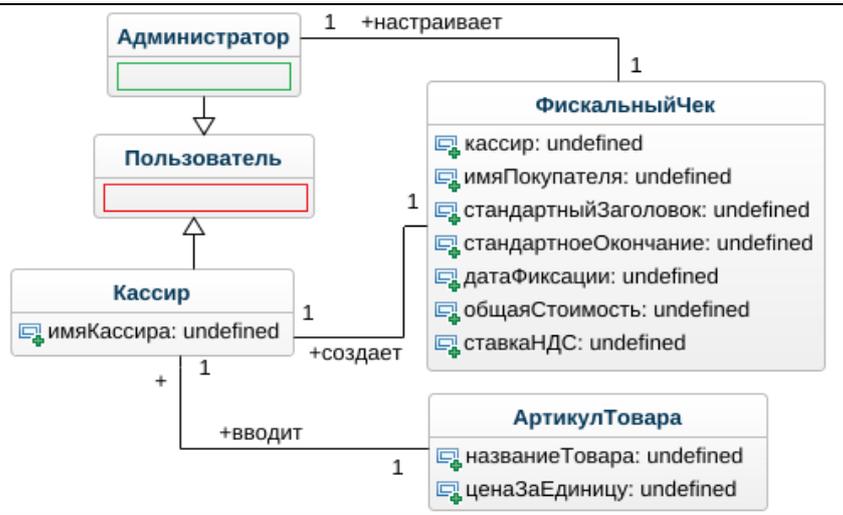
Таблица 6.1 – Ошибки в диаграммах понятий

Номер и текст ошибки	Пример с ошибкой
1	2
Лексические ошибки	
6.1 Использование элементов, которые не являются лексемами для диаграммы понятий	
Элемент «Актор» не входят в список лексем, разрешенный стандартом UML для CD	
6.2 Неверное имя понятия	
Имя понятия не должно выражать действие. Имя понятия должно быть представлено именем существительным, начинаться с заглавной буквы и не содержать символы, недопустимые в идентификаторах. Рекомендуется использование венгерской нотации	
6.3 Неверное имя атрибута понятия	
Имя атрибута понятия должно быть представлено именем существительным, начинаться с прописной буквы и не содержать символы, недопустимые в идентификаторах. Рекомендуется использование венгерской нотации. Имя атрибута понятия не должно выражать действие	

Продолжение таблицы 6.1

1	2	
6.4 Указание части класса у понятия		
<p>Элементами диаграммы понятий являются «понятия», а не классы. Указание части класса, отличной от установленной по умолчанию (+), является избыточным</p>		
6.5 Указание типа данных у атрибута понятия		
<p>Элементами диаграммы понятий являются «понятия», а не классы. Указание типа данных у атрибута понятия, отличного от установленного по умолчанию, является избыточным</p>		
6.6 Использование неверного элемента CD для обозначения ассоциации		
<p>Подмена ассоциации элементом «Зависимость», который является лексемой диаграммы классов, но не может быть частью диаграммы понятий</p>		
6.7 Неверное указание ограничения		
<p>Текст ограничения должен быть заключен в фигурные скобки</p>		
<p style="text-align: center;">Синтаксические ошибки</p>		
6.8 Использование имени роли ассоциации вместо имени ассоциации		
<p>Имя роли ассоциации – это имя существительное, оно является названием роли, в которой выступает понятие в этой ассоциации. Имя ассоциации всегда выражает действие. Поэтому имя роли ассоциации не может быть именем ассоциации</p>		

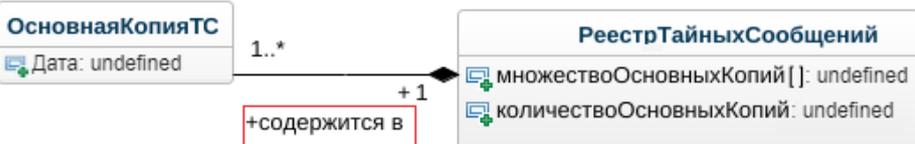
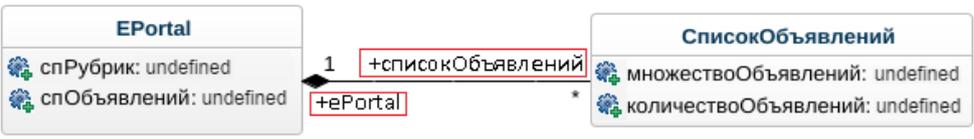
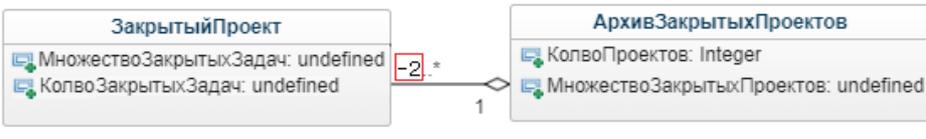
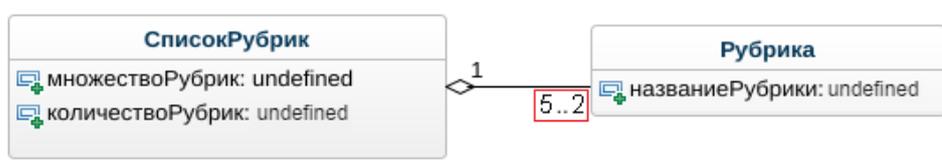
Продолжение таблицы 6.1

1	2
	
	
6.9 Понятие не содержит ни один атрибут	<p>У ИС атрибутом может быть ее название, а у системы помощи – название файла помощи или тип такого файла</p>
	
	
	<p>В случае в) понятия-потомки не имеют атрибутов, но отличаются принадлежностью к различным понятиям-контейнерам. В случае г) понятие-потомок Администратор не имеет атрибутов и отличается от понятия-потомка Кассир участием в ассоциации, отличной от ассоциации Кассира. У понятия Пользователь атрибутами могут быть логин и пароль</p>

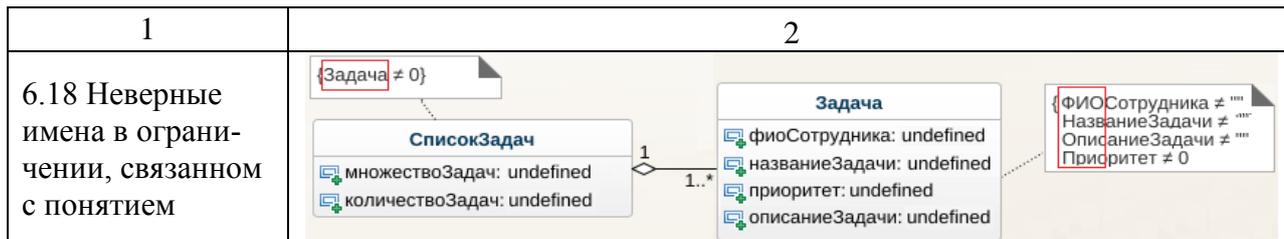
Продолжение таблицы 6.1

1	2	
<p>6.10 Существует понятие, не имеющее никаких ассоциаций</p>		
<p>У каждого понятия должна существовать хотя бы одна ассоциация с другими понятиями</p>		
<p>6.11 Отсутствие атрибута для реализации агрегации / композиции</p>		
<p>Понятие-контейнер должно иметь атрибут для содержимого этого контейнера</p>		

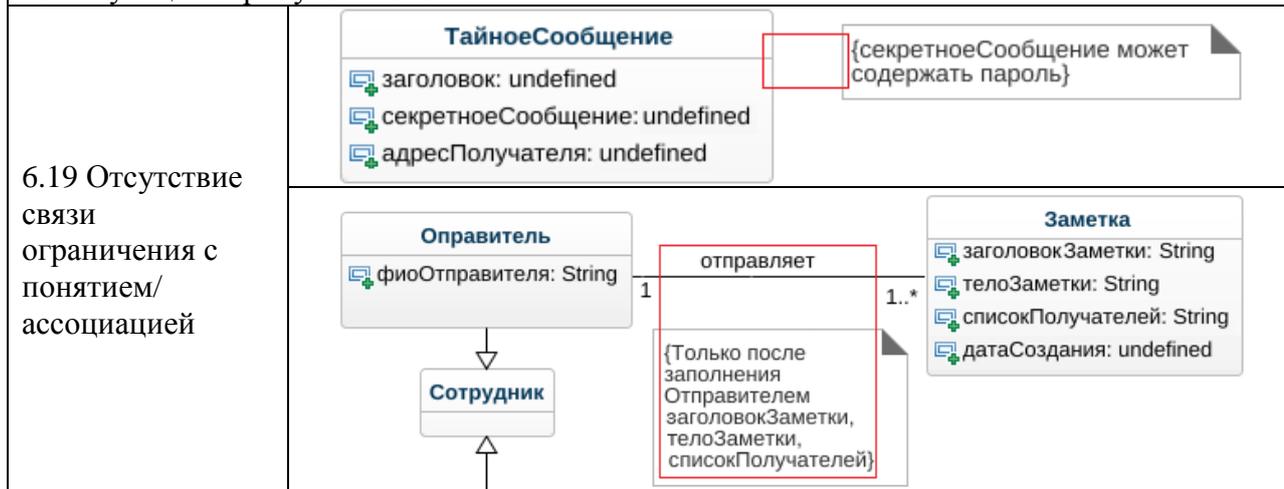
Продолжение таблицы 6.1

1	2
6.12 Подмена агрегации обобщением	
<p>Отношение «часть-целое» изображается символом агрегации / композиции. Символ обобщения соответствует отношению «супертип-подтип» («предок-потомок»)</p>	
6.13 Лишнее имя у символа агрегации / композиции	
<p>Для именованной агрегации / композиции использовано имя роли. Отношения агрегации и композиции изображаются соответствующими лексемами диаграммы, поэтому их именование не требуется</p>	
6.14 Избыточное имя роли агрегации / композиции	
<p>Отношения агрегации и композиции изображаются соответствующими лексемами диаграммы, поэтому именование их ролей не требуется</p>	
6.15 Отсутствие кратности у роли агрегации / композиции	
<p>При использовании отношения «часть-целое» требуется явно указать, сколько частей содержит целое – одну или несколько. Это выполняется указанием значений кратности ролей</p>	
6.16 Использование при указании кратности роли неверного значения	
<p>При указании кратности роли все числа должны быть целыми неотрицательными</p>	
6.17 Использование при указании кратности роли неверного диапазона значений	
<p>При указании кратности роли в виде диапазона (промежутка) он должен начинаться с меньшего значения</p>	

Продолжение таблицы 6.1

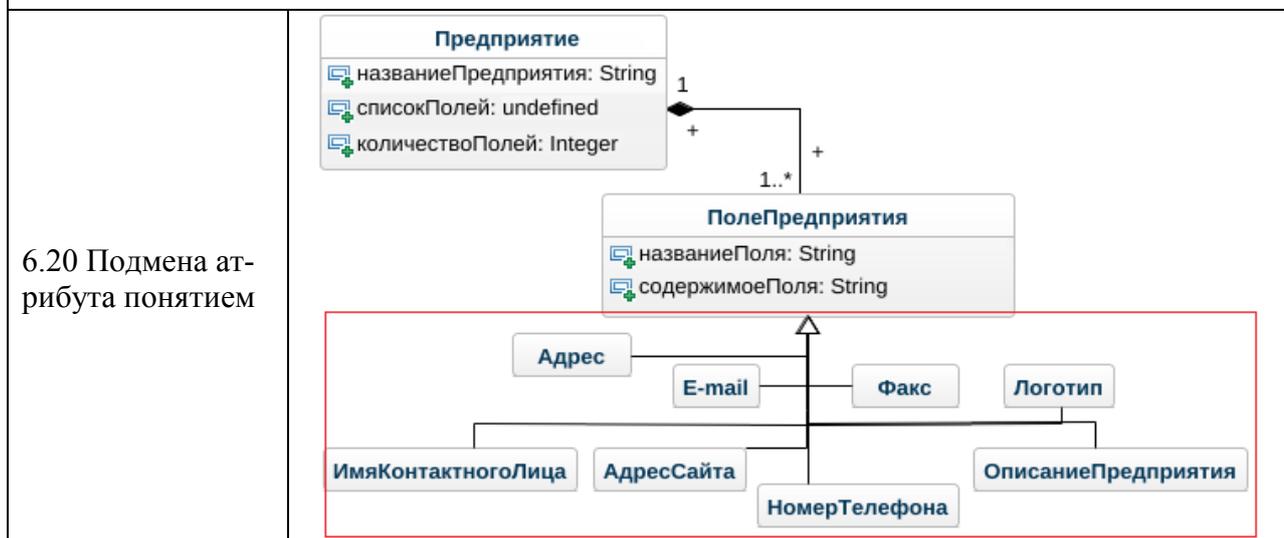


Имена в тексте ограничения, связанного с понятием, должны совпадать с именами соответствующих атрибутов этого понятия



Каждое ограничение должно быть связано с соответствующим понятием или ассоциацией. Если текст в элементе «Comment» представляет собой комментарий, то заключать его в фигурные скобки не следует

Семантические ошибки



Все «подтипы» (понятия-потомки) иерархии обобщения являются атрибутами понятия Предприятие. Поэтому лишними являются понятие ПолеПредприятия и его понятия-потомки, а также отношения обобщение и композиция. Такая диаграмма свидетельствует о том, что автор мыслит на уровне данных, а не на уровне модели

Продолжение таблицы 6.1

1	2
6.21 Лишний атрибут понятия	

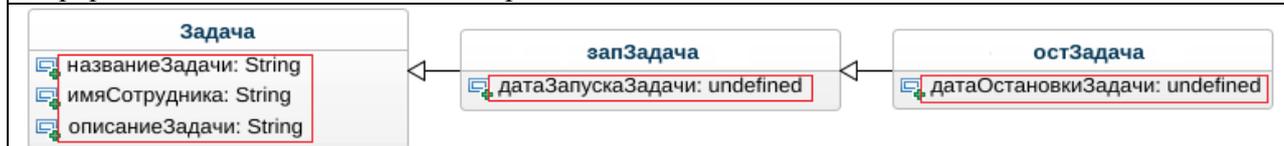
Для идентификации понятий используются их уникальные имена. Такая диаграмма свидетельствует о том, что автор мыслит на уровне данных, а не на уровне модели

6.22 Дублирование атрибутов у понятий супертипа и подтипа	
---	--

Ответы на вопросы «Что имеют общее?» и «Чем отличаются?» о понятиях Задача, запЗадача и остЗадача позволяют выявить их общие атрибуты: названиеЗадачи, имяСотрудника и описаниеЗадачи. Атрибутами, которыми отличаются эти понятия, являются датаЗапускаЗадачи и датаОстановкиЗадачи. Поэтому у супертипа Задача должны быть эти три атрибута. Остальные атрибуты должны быть распределены между подтипами запЗадача и остЗадача.

Ответы на вопросы «Что имеют общее?» и «Чем отличаются?» о понятиях-подтипах запЗадача и остЗадача позволяют выявить общий атрибут – датаЗапускаЗадачи, т.к. выполнение и запущенной, и остановленной задачи должно когда-то начинаться. Поэтому понятие запЗадача является супертипом для понятия остЗадача и должно иметь атрибут датаЗапускаЗадачи. У подтипа остЗадача должен быть атрибут датаОстановкиЗадачи, который позволяет отличать понятие остЗадача от супертипа запЗадача.

Иерархия наследования для этой ПрО должна иметь вид:



Также в именах подтипов допущена ошибка 6.2

Продолжение таблицы 6.1

1	2
6.23 Дублирование атрибутов у подтипов (понятий-потомков)	
<p>Ответы на вопросы «Что имеют общее?» и «Чем отличаются?» понятия Пользователь, Кассир, КонтролерКассира и Администратор позволяют выявить их общие атрибуты: фамилия и имя. Поэтому эти два атрибута должны быть у супертипа Пользователь. Подтипы Кассир, КонтролерКассира и Администратор различаются между собой поведением. Также каждый из них имеет отношение хотя бы с одним понятием, которое не связано с остальными понятиями-потомками. Подобная ситуация представлена в ошибке 6.24</p>	
6.24 Отсутствие обобщения	
<p>Ответы на вопросы «Что имеют общее?» и «Чем отличаются?» понятия Товар и ТоварСНДС позволяют выявить отношение обобщения</p>	
6.25 Неверное направление у обобщения	
<p>При таком направлении обобщения понятие Предприятие кроме собственных атрибутов названиеПредприятия, адрес и т.д., получает атрибуты подтверждениеАдминистратора и отклонениеАдминистратора, что противоречит описанию ПрО</p>	
6.26 Лишняя ассоциация между понятиями «супер-тип» и «подтип»	
<p>При наличии обобщения между двумя понятиями (предок-потомок) других структурных отношений между ними не должно быть</p>	

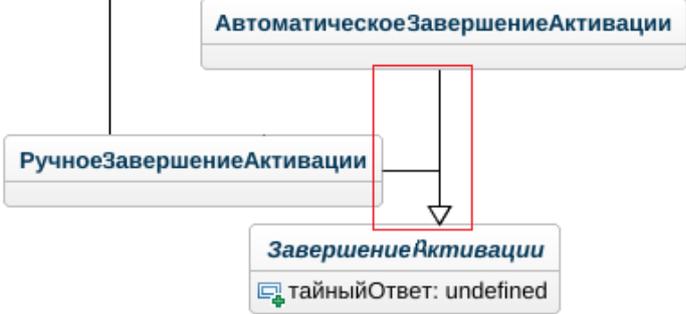
Продолжение таблицы 6.1

1	2
<p>6.27 Лишняя ассоциация между понятиями-потомками</p>	
<p>В иерархии обобщения понятия «подтипа» могут иметь структурные отношения только с понятием «супертипа»</p>	
<p>6.28 Лишние значения у свойств разбиения при многомерной множественной классификации</p>	<p>Свойства разбиения имеют значения по умолчанию «complete» и «disjoint», их явное указание избыточно</p>
<p>6.29 Неверное направление у агрегации / композиции</p>	
<p>При указании отношения «часть-целое» ромб должен быть размещен на стороне «целое»</p>	
<p>6.30 Неверная кратность агрегации (должна быть «один-ко-многим»)</p> <p>Если атрибутом производного понятия-контейнера («целое») в агрегации / композиции является базовое понятие-содержимое контейнера («часть»), то кратность роли «части» должна быть больше единицы. Если атрибутом производного понятия-контейнера («целое») в агрегации / композиции «частью» является базовое понятие-содержимое контейнера, то кратность роли «целого» должна быть равна 1</p>	

Продолжение таблицы 6.1

1	2
<p>6.31 Неверная кратность агрегации / композиции (должна быть «один-к-одному»)</p>	
<p>Если атрибутом производного понятия («целое») в агрегации / композиции «частью» является базовое понятие-контейнер, то кратности ролей «целого» и «части» должны быть равны единице</p>	
<p>6.32 «Цикл» в агрегации / композиции между понятиями</p>	
<p>«Целое» (понятие СписокОбъявлений) не может быть одновременно «частью» самого себя (понятие Объявление)</p>	
<p>6.33 Существование между парой понятий отношений обобщение и агрегация/ композиция одновременно</p>	
<p>Существовать между парой понятий оба отношения (обобщение и агрегация) одновременно не могут, т.к. обобщение – это иерархия классов (категорий, понятий), а агрегация – это иерархия объектов (экземпляров понятий). Поэтому эти два вида ассоциаций являются взаимоисключающими</p>	
<p>6.34 Неверная связь ограничения с понятием</p>	
<p>В ограничении, связанном с производным понятием («целое»), использованы имена атрибутов базового понятия («части»). Поэтому ограничение должно быть связано с «частью» (понятие Задача)</p>	

Продолжение таблицы 6.1

1	2
<p>6.35 Использование в ограничении имени понятия, а не имени его атрибута</p>	
<p>Операция, использованная в ограничении, не может быть применена к понятию, это операция для его атрибута</p>	
<p>Ошибки размещения</p>	
<p>6.36 Символы нескольких обобщений между понятиями накладываются</p>	
<p>Эстетическая аномалия: может возникать при использовании обобщения между понятиями. Возможны наложения фрагментов нескольких символов обобщения</p>	

7. Создание SD

Как и другие, рассмотренные ранее виды диаграмм (прецедентов и понятий), диаграммы последовательностей могут иметь различный уровень детализации. В SD с Low LoD системные сообщения и сообщения между объектами имеют только метку сообщения. В диаграммах последовательностей с High LoD эта метка представляет собой имя сообщения со списком параметров.

Представление SD с High LoD является целью темы 7. Это связано с тем, что на этапе проектирования будут учитываться особенности потенциального языка ООП, определенного для этапа реализации. Для создания диаграмм последовательностей следует использовать сервис genMyModel.

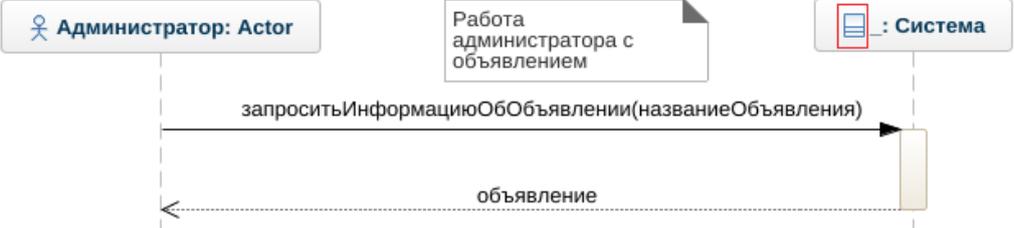
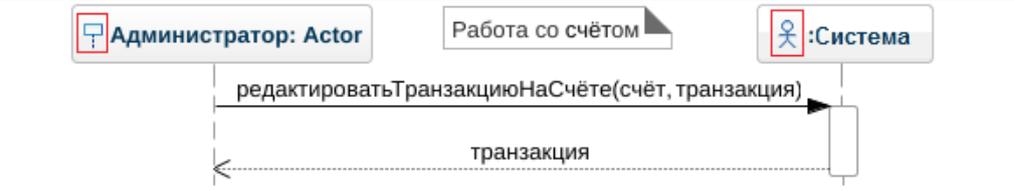
Лексическими являются ошибки, которые нарушают правила создания и использования лексем подмножества UML для построения диаграмм последовательностей с High LoD – допустимых графических образов субъектов (система, акторы, объекты, фокусы управления, сообщения и список их параметров) и их имен.

Синтаксические ошибки возникают при отсутствии субъектов, неверном установлении связей между субъектами и/или их размещении по отношению друг к другу, а также при нарушении принципа идентификации.

К семантическим ошибкам приводит неверное понимание уровня моделирования («скатывание» с уровня модели на уровень данных) или неверная передача данных с помощью параметров сообщений или значений, возвращаемых ответными сообщениями.

В таблице 7.1 представлены типичные ошибки, допускаемые студентами при создании диаграмм последовательностей с High LoD.

Таблица 7.1 – Ошибки в SD с высоким уровнем детализации

Номер и текст ошибки	Пример с ошибкой
1	2
Лексические ошибки	
7.1 Неверная пиктограмма объекта	
	

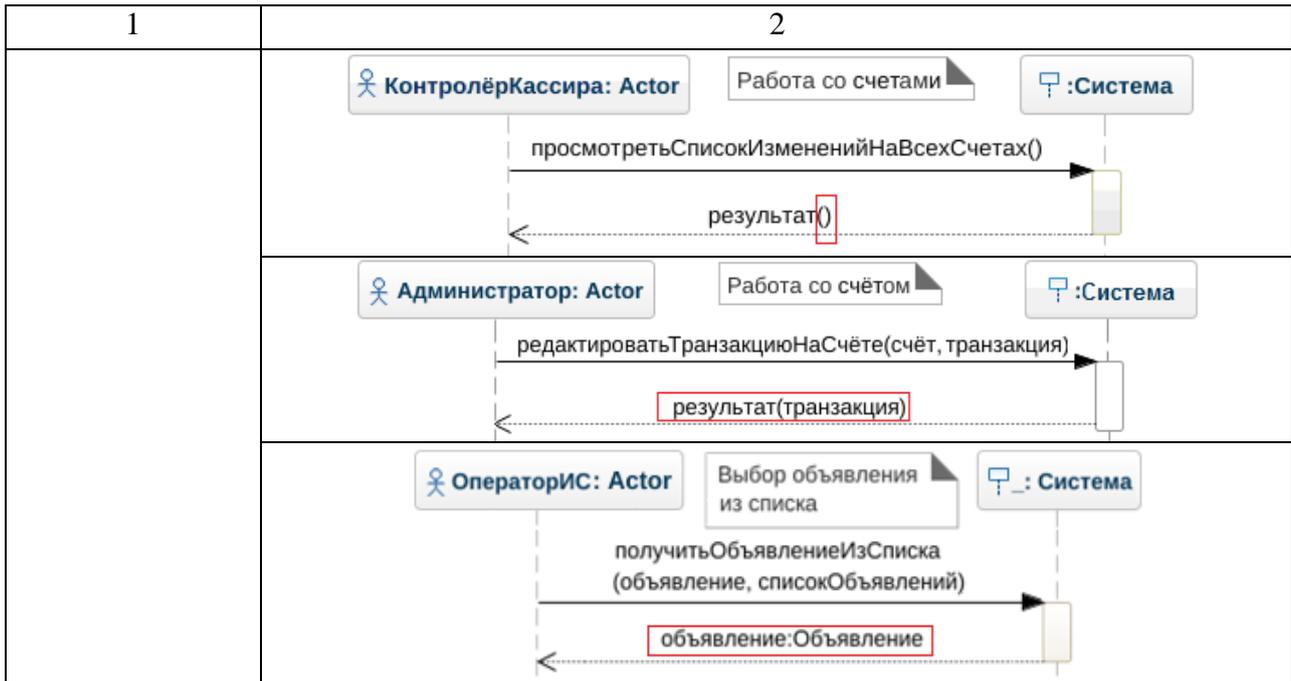
Продолжение таблицы 7.1

1	2
<p>Для обозначения системы используется пиктограмма объекта с вертикальной линией жизни. Однако такое образ не подходит для актора, имя которого должно сопровождаться пиктограммой, принятой для их изображения в UCD. И наоборот, пиктограмма актора не может быть использована как образ системы</p>	
<p>7.2 Неверное имя объекта</p>	
	<p>Имя объекта должно обязательно содержать имя класса, указанное после символа двоеточия. У актора именем класса является «Actor», у системы – «Система» или «System». Имя объекта и имя класса, разделенные двоеточием, должны быть представлены именами существительными, начинаться с заглавной буквы и не содержать символы, недопустимые в идентификаторах. Имя объекта в отличие от имени класса не является обязательным</p>
<p>7.3 Неверное имя системного сообщения</p>	

Продолжение таблицы 7.1

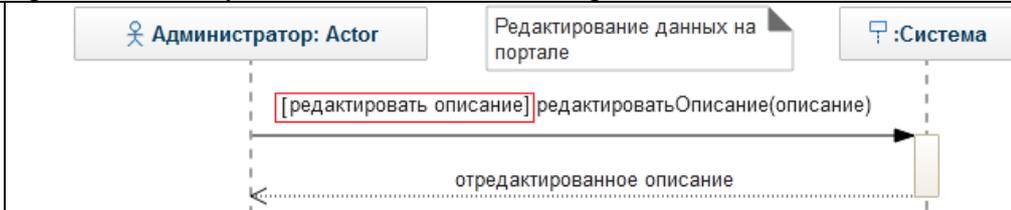
1	2
<p>Имя системного сообщения должно быть представлено глаголом в повелительном наклонении, начинаться с прописной буквы и не должно содержать символы, недопустимые в идентификаторах. Рекомендуется использование венгерской нотации</p>	
<p>7.4 Неверное имя параметра системного сообщения</p>	
<p>Имя параметра системного сообщения должно быть представлено именем существительным, начинаться с заглавной буквы и не содержать символы, недопустимые в идентификаторах. Рекомендуется использование венгерской нотации. Также в модели присутствует ошибка размещения, полученная при экспорте графического файла: наложение текста системного сообщения на его стрелку</p>	
<p>7.5 Отсутствие ответного сообщения у синхронного системного сообщения</p>	
<p>В случае отправки синхронного системного сообщения его отправитель дожидается завершения обработки этого сообщения. Подтверждением завершения обработки является получение отправителем соответствующего ответного сообщения. Если ответное сообщение не указано, то отправленное СС должно быть асинхронным</p>	
<p>7.6 Неверное имя данно-го/объекта на ответном сообщении синхронного системного сообщения</p>	

Продолжение таблицы 7.1



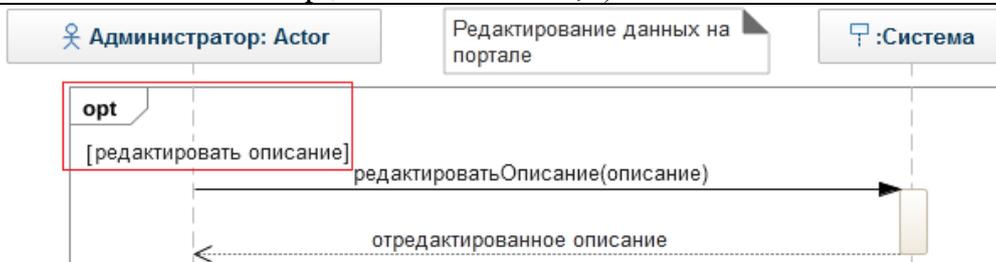
Подтверждением завершения обработки синхронного системного сообщения является получение отправителем соответствующего ответного сообщения. Ответное сообщение всегда возвращает отправителю управление потоком сообщений. При необходимости ответное сообщение может возвращать данные (-ые) /объект (-ы). В этом случае на стрелке ответного сообщения указывается имя или список имен этих данных/объектов, но не тип данных. Имя данного/объекта на ответном сообщении должно быть представлено именем существительным, начинаться с прописной буквы и не содержать символы, недопустимые в идентификаторах. Рекомендуется использование венгерской нотации

7.7 Неверное условное системное сообщение



а)

В ошибке 7.7, а) показана попытка создания условного системного сообщения. В GenMyModel используется нотация MSC, поэтому условное сообщение должно быть оформлено с помощью блока opt, как в ошибке 7.7, б):

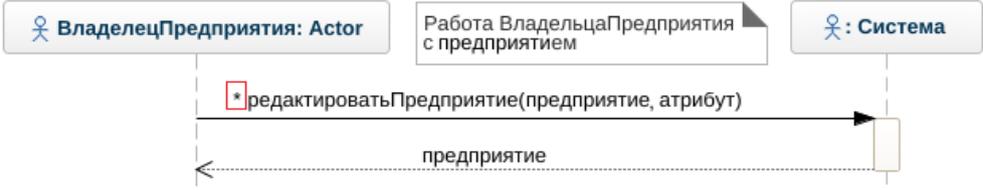
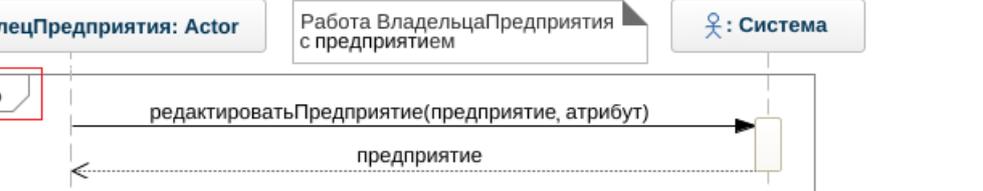
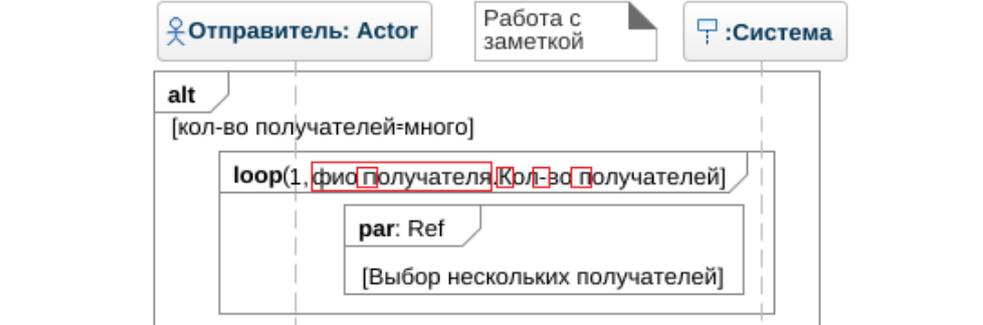


б)

Продолжение таблицы 7.1

1	2
<p>7.8 Неверное обозначение блока opt условного системного сообщения</p>	
<p>Наименование блока opt не должно сопровождаться никакими символами (двоеточием, текстом условия). Условие указывается в теле этого блока в квадратных скобках</p>	
<p>7.9 Неверное условие условного системного сообщения</p>	
<p>Условие условного системного сообщения должно быть заключено в квадратные скобки</p>	
<p>7.10 Неверное условие рефлексивного сообщения</p>	
<p>Условие рефлексивного сообщения должно быть заключено в квадратные скобки</p>	
<p>7.11 Неверное обозначение блока alt альтернативных системных сообщений</p>	
<p>Наименование блока alt не должно сопровождаться никакими символами</p>	

Продолжение таблицы 7.1

1	2
7.12 Неверное повторяющееся системное сообщение	 <p style="text-align: center;">а)</p>
<p>В ошибке 7.12, а) показана попытка создания повторяющегося системного сообщения; в GenMyModel используется нотация MSC, поэтому повторяющееся сообщение должно быть оформлено с помощью блока loop как в 7.12, б). Для бесконечного цикла достаточно указать только название блока – «loop»</p>	
	 <p style="text-align: center;">б)</p>
7.13 Неверное имя атрибута в границе диапазона цикла повторяющегося системного сообщения	
<p>Имена объектов и их атрибутов, использованные для указания значения в границе диапазона цикла повторяющегося системного сообщения, должны быть представлены именами существительными, начинаться с прописной буквы и не должны содержать символы, недопустимые в идентификаторах. Рекомендуется использование венгерской нотации</p>	
Синтаксические ошибки	
7.14 Отсутствие названия прецедента	
<p>Диаграмма последовательностей – это формальное представление некоторого сценария прецедента. Поэтому для SD необходимо указывать название соответствующего прецедента</p>	
7.15 Неверное размещение названия прецедента	
<p>Название прецедента должно быть внутри диаграммы последовательностей</p>	

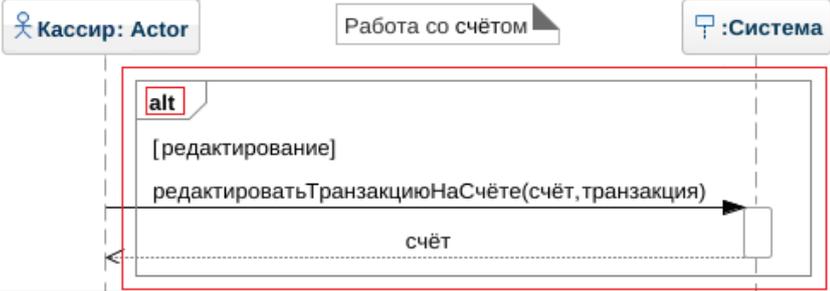
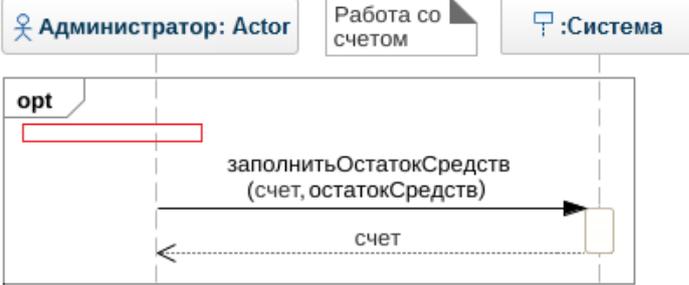
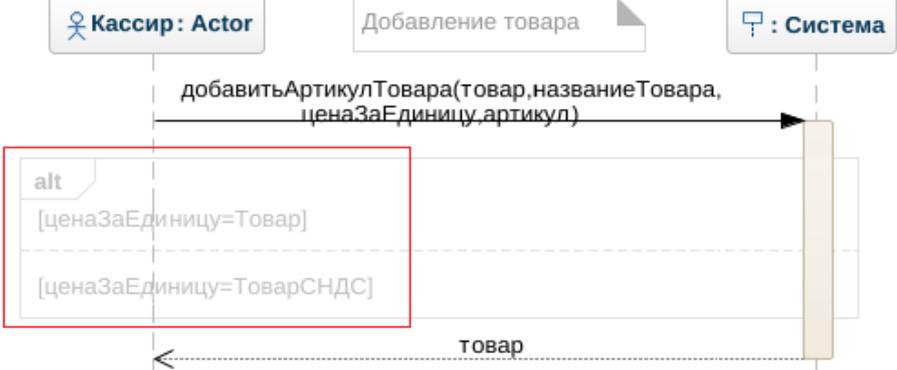
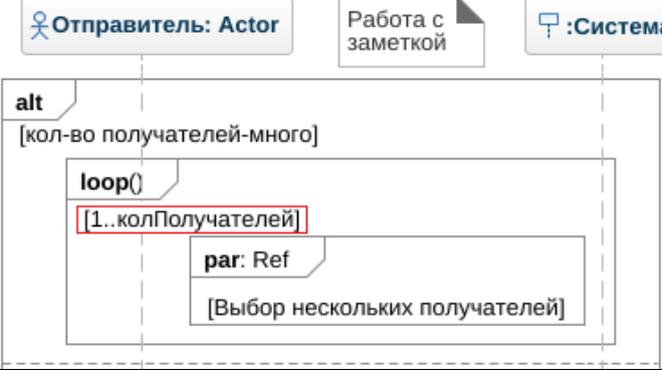
Продолжение таблицы 7.1

1	2
7.16 Отсутствие списка параметров у системного сообщения	<pre> sequenceDiagram actor Actor as Отправитель: Actor participant System as :Система Actor->>System: отменитьОтправкуЗаметки System-->>Actor: отправка отменена </pre>
<p>При выполнении задания этой темы должны быть разработаны диаграммы последовательностей с высоким уровнем детализации</p>	
7.17 Отсутствие списка параметров у рефлексивного сообщения	<pre> sequenceDiagram actor Actor as Отправитель: Actor participant System as :Система Actor->>System: создатьНовуюЗаметку(заголовок) activate System System->>System: создатьЗаметку deactivate System System-->>Actor: заметка </pre>
<p>При разработке диаграммы последовательностей с High LoD метка рефлексивного сообщения должна иметь кроме имени сообщения соответствующий список параметров</p>	
7.18 Имена системных сообщений должны быть уникальными	<pre> sequenceDiagram actor Actor as ВладелецПредприятия: Actor participant System as : Система alt [не требуется оплата] Actor->>System: изменитьПакет (предприятие, пакет) System-->>Actor: пакет else [требуется оплата] Actor->>System: изменитьПакет (предприятие, пакет) System-->>Actor: суммаОплаты end </pre>
<p>Нарушение принципа идентификации. Системное сообщение – это запрос актора к некоторой функции системы. Невозможно на один и тот же запрос получать ответы от различных функций</p>	
7.19 Имя системного сообщения и рефлексивного сообщения не должны совпадать	<pre> sequenceDiagram actor Actor as Отправитель: Actor participant System as :Система Actor->>System: создатьЗаметку(заголовок) activate System System->>System: создатьЗаметку(заголовок) deactivate System System-->>Actor: заметка </pre>
<p>Нарушение принципа идентификации. Системное сообщение соответствует внешнему событию, которое может быть распознано системой и далее ею обработано. Рефлексивное сообщение соответствует действиям по обработке распознанного системного сообщения внутри системы. В общем случае и системное сообщение, и рефлексивное сообщение при реализации класса в конкретном языке ООП будут представлены функциями-членами класса / методами, которые могут быть перегружены. Если одноименные сообщения в SD отличаются количеством параметров, то перегрузка сообщений правильная (ошибка 7.11). Но на этапе анализа в SD невозможна типизация параметров сообщений, поэтому гарантировать правильность перегрузки сообщений с одинаковым списком параметров невозможно</p>	

Продолжение таблицы 7.1

1	2
<p>7.20 Лишний фокус управления у актора</p>	
<p>Фокус управления иллюстрирует действия объекта по обработке сообщения. Актору, который является отправителем системных сообщений, фокус управления не требуется</p>	
<p>7.21 Неверное создание объекта / актора</p>	
<p>Для создания объекта необходимо отправить сообщение, а не только указать имя создаваемого объекта</p>	
<p>7.22 Использование синхронного системного сообщения вместо ответного</p>	
<p>Системное сообщение – это входящее сообщение, сгенерированное актором для системы и переданное через ее границу. Поэтому система не может быть отправителем системных сообщений</p>	
<p>7.23 Лишняя информация у объекта на ответном сообщении</p>	
<p>Ответное сообщение синхронного системного сообщения может возвращать данное (или их список) / объект (или их список). Если в SD должен быть возвращен: а) объект «отправленнаяЗаметка», то перечисленные в круглых скобках имена являются именами атрибутов этого объекта, поэтому их указание является ошибкой, б) список данных «заголовокЗаметки», «телоЗаметки», ... «датаОтправкиЗаметки», то указание имени объекта «отправленнаяЗаметка» является ошибкой</p>	

Продолжение таблицы 7.1

1	2
<p>7.24 Использо- вание блока alt вме- сто opt для услов- ного системного сообщения</p>	
<p>В нотации MSC блок opt используется для описания необязательного шага в рабочем процессе. Блок alt используется для описания альтернативных сценариев рабочего процесса: будет выполнен только один из вариантов</p>	
<p>7.25 Отсутствие условия в блоке opt</p>	
<p>В нотации MSC блок opt используется для описания необязательного шага в рабочем процессе, поэтому указание условия осуществления этого шага является обязательным</p>	
<p>7.26 Отсутствие сообщений в блоке alt</p>	
<p>Ветвление процесса с помощью блока alt предусмотрено, но сообщения по соответствующим ветвям не указаны</p>	
<p>7.27 Неверное задание диапа- зона цикла в блоке loop</p>	
<p>Для цикла с известным числом повторения после loop в круглых скобках следует указать количество шагов цикла или нижнюю и верхнюю границы диапазона через запятую</p>	

Продолжение таблицы 7.1

1	2
Семантические ошибки	
<p>7.28 Неверное имя системного сообщения для создания объекта</p>	
<p>Имя системного сообщения отражает действия, которые должны быть выполнены в ответ на это сообщение. Невозможно «предоставить...» объект, которого нет, но при этом моделировать его создание. Поэтому системное сообщение о создании объекта должно содержать в имени действие «создать...»</p>	
<p>7.29 Отсутствие имени у созданного объекта</p>	
<p>Если системное сообщение о создании объекта возвращает на ответном сообщении имя созданного объекта, то созданный объект не должен быть анонимным</p>	
<p>7.30 Недопустимое уничтожение актора</p>	
<p>Актор генерирует входящее сообщение для системы. Поэтому он должен существовать в течение жизни этой системы</p>	
<p>7.31 Пустой список параметров системного сообщения</p>	
<p>Имя системного сообщения отражает действия о заполнении данными карточки предприятия, но значения этих данных не поступают от актора параметрами сообщения</p>	

Продолжение таблицы 7.1

1	2
<p>7.32 Неверное системное сообщение</p>	<pre> sequenceDiagram actor Admin as Администратор: Actor participant System as :Система note over Admin, System: Работа администратора с предприятием activate System Admin->>System: редактироватьПредприятие(предприятие, название, адрес, телефон, факс, e-mail, контактноеЛицо, сайт, описание, логотип) activate System System-->>Admin: предприятие deactivate System </pre>
<p>Имя системного сообщения отражает действия о редактировании атрибутов предприятия. Но указанный список параметров указывает, что обязательно должны быть изменены все атрибуты, для которых параметрами переданы соответствующие значения. По сценарию взаимодействия с системой в рамках подобного прецедента должна быть предусмотрена модификация одного или нескольких атрибутов с учетом потребностей актора. Для устранения ошибки это системное сообщение должно быть представлено последовательностью условных сообщений для редактирования каждого из атрибутов предприятия</p>	
<p>7.33 Недостаточно параметров у системного сообщения</p>	<pre> sequenceDiagram actor Cashier as Кассир: Actor participant System as :Система note over Cashier, System: Работа со счетом activate System Cashier->>System: редактироватьТранзакциюНаСчете(счёт, транзакция) activate System System-->>Cashier: счёт deactivate System </pre>
<p>Имя системного сообщения отражает действия о редактировании счета, но данные для этих действий не поступают от актора параметрами сообщения</p>	
<p>7.34 Лишнее системное сообщение</p>	<pre> sequenceDiagram actor User as Пользователь: Actor participant System as :Система note over User, System: Создание и публикация карточки предприятия activate System User->>System: сохранитьКарточкуПредприятия() activate System System-->>User: карточка предприятия deactivate System actor User as Пользователь: Actor participant System as :Система note over User, System: Работа с договором activate System User->>System: выйтиИзРежимаИзменения(измененныйДоговор) activate System System-->>User: результат закрытия режима изменения deactivate System actor Admin as Администратор: Actor participant System as :Система note over Admin, System: Работа со счетом activate System Admin->>System: закончитьРедактирование() activate System System-->>Admin: deactivate System </pre>
<p>Имена системных сообщений не отражают действия, которые могут привести к изменениям в объектно-ориентированной системе на уровне модели. Ошибка, связанная с действием сохранения (уровень данных), рассмотрена в ошибке 4.40, ж). Действие изменения режима работы ИС (уровень данных / уровень представления) является ошибкой</p>	

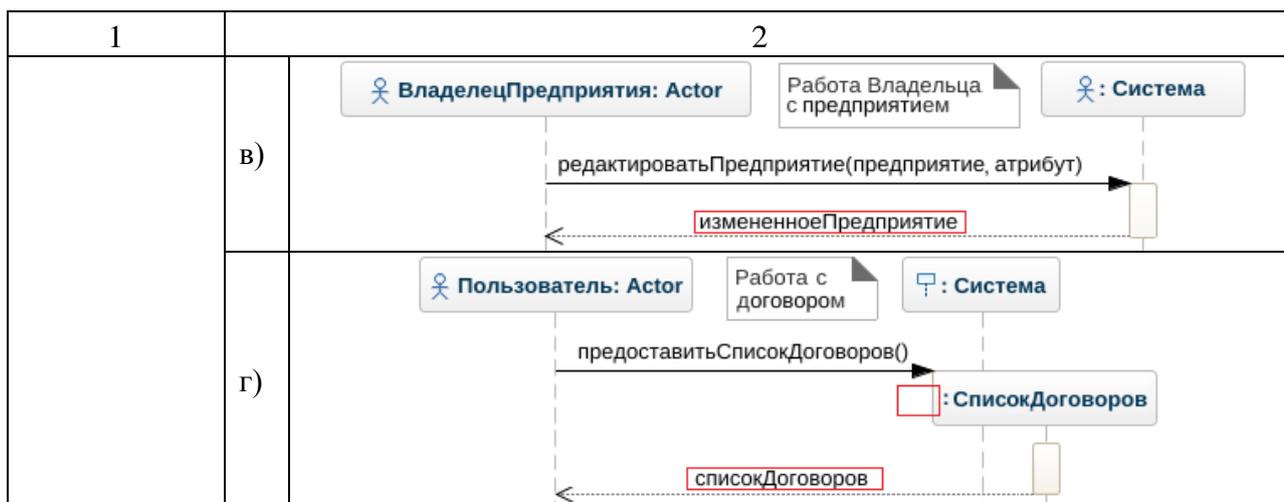
Продолжение таблицы 7.1

1	2
<p>7.35 Лишнее рефлексивное сообщение</p>	
<p>Ошибка, связанная с действием сохранения (уровень данных), рассмотрена в ошибке 4.40, ж)</p>	
<p>7.36 Отсутствие ответного сообщения при наличии рефлексивного сообщения</p>	
<p>В результате обработки системного сообщения в системе создается объекта подтипа (понятие-потомок основнаяКопия) на основе объекта супертипа (понятия-предка тайное сообщение). Описание соответствующей ПрО содержит информацию, что «..управлять основной копией может любой получатель этого секретного сообщения..». Но в этой SD «основнаяКопия» после создания доступна только в системе. Поэтому эта диаграмма последовательностей может иметь вид, представленный в 7.36 ,б):</p>	
<p>7.37 Неизвестные значения параметров рефлексивного сообщения</p>	
<p>Рефлексивное сообщение использует значения параметров, которые не были заданы актором и не были предварительно получены в самой системе</p>	

Продолжение таблицы 7.1

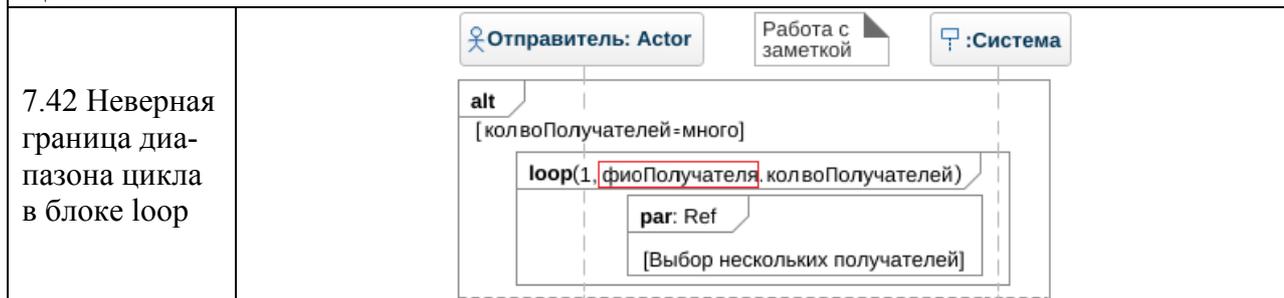
1	2
<p>7.38 Отсутствие рефлексивного сообщения при наличии условия его отправки</p>	
<p>Внутри системы указано условие отправки рефлексивного сообщения, но соответствующее сообщение отсутствует</p>	
<p>7.39 Наличие у рефлексивного сообщения на ответном условии вместо данного</p>	
<p>Внутри системы указано условие отправки ответного сообщения для рефлексивного вместо указания данного, которое может быть возвращено. Ошибка, связанная с действием сохранения (уровень данных), рассмотрена в ошибке 4.40, ж)</p>	
<p>7.40 Отсутствие создания понятия-потомка</p>	
<p>Отсутствует создание объекта подтипа (понятия-потомка отправленнаяЗаметка) на основе объекта супертипа (понятия-предка заметка). При этом имя понятия-потомка используется как параметр сообщения и как данное у ответных сообщений. При таком уровне детализации создание понятия-потомка необходимо указать явно до его первого использования</p>	
<p>7.41 Неверная семантика данного на ответном сообщении</p>	<p>а)</p> <p>б)</p>

Продолжение таблицы 7.1



Ответное сообщение может использоваться для передачи отправителю результата обработки его сообщения. Неверными результатами являются:

- в а) объект уровня представления вместо объекта уровня модели,
- в б) действие вместо данного или объекта,
- в в) неизвестный объект вместо исходного объекта, переданного параметром, у которого при обработке системного сообщения было изменено состояние,
- в г) неизвестный объект, имя которого не было указано при обработке системного сообщения



У объекта с именем «фиоПолучателя» не может быть числового атрибута (количество)



По сценарию в этом фрагменте SD при выборе получателя по его персональным данным на каждом шаге цикла из системы должен быть получен объект списокСотрудников. Это является ошибкой: последнее действие следует выполнить один раз до цикла loop

Продолжение таблицы 7.1

1	2
<p>7.44 Недопустимое создание атрибутов объекта</p>	
<p>Атрибуты объекта – это его статическая часть, которая описывает состояние объекта. Невозможно атрибуты объекта создавать, их значения можно использовать или устанавливать, соответственно наблюдая или изменяя состояние этого объекта. В этой SD параметром передано значение «заголовокЗаметки», но в рекурсивном сообщении выполнена неверная попытка создать объект с таким именем. На ответном сообщении указано имя неизвестного объекта «созданнаяЗаметка» и в скобках неверно указано либо исходное значение «заголовокЗаметки», либо объект с таким же именем, который должен был быть создан системой. Аналогичная цепочка ошибок связана и с именем параметра «телоЗаметки» системного сообщения</p>	
<p>Ошибки размещения</p>	
<p>7.45 Текст системного сообщения пересекается линией стрелки этого сообщения</p>	
<p>Эстетическая аномалия. Она также присутствует в ошибках 7.26, 7.43, 8.1, 8.2</p>	
<p>7.46 Символы нескольких фокусов управления накладываются</p>	
<p>Эстетическая аномалия</p>	

8. Описание системных операций

В теме 8 завершается работа на этапе анализа и готовится информация для перехода от этапа анализа к этапу проектирования ИС. Поэтому важной является согласованность описания системных операций с другими моделями этапа анализа:

- диаграммой последовательностей,
- диаграммой понятий.

Системное сообщение SD, указанное в сценарии прецедента, с точки зрения кодирования на языке объектно-ориентированного программирования представляет собой вызов функции:

- с типом результата `void`, если ответное сообщение возвращает только управление (поток событий),
- с типом результата, отличным от `void`, если ответное сообщение дополнительно возвращает данные / объект.

Описание системной операции, которая соответствует какому-то системному сообщению, представляет собой спецификацию тела этой функции с указанием типов данных всех объектов, которые используются при реализации в системе этой операции.

Поэтому должно поддерживаться полное соответствие:

- количества системных сообщений согласно классификации прецедентов и количества описаний системных сообщений,
- имени системного сообщения в SD и имени системной операции,
- списка параметров (количества параметров и порядка их следования) у системного сообщения в SD и списка параметров системной операции.

Нарушение этого соответствия является ошибкой.

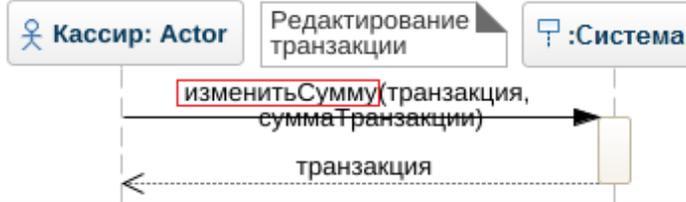
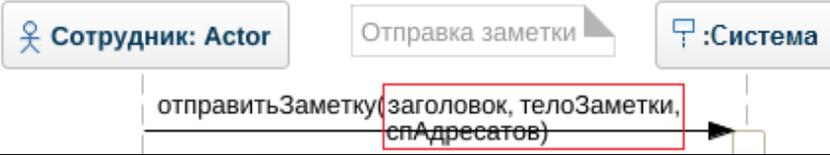
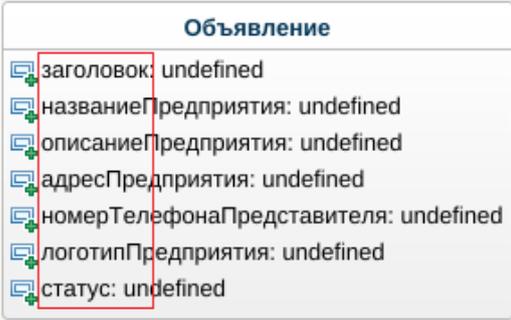
Имена понятий и их атрибутов, которые были зафиксированы в результате концептуального моделирования, соответствуют именам классов и их атрибутов в описании системных операций. Нарушение этого соответствия является ошибкой.

В объектно-ориентированной системе данными являются значения атрибутов объектов системы. Изменения на этом уровне сводятся к созданию/уничтожению экземпляров (объектов), формированию/разрыву связей между экземплярами и изменению атрибутов. В разделе постусловий рекомендуется «...каждое изменение явно относить к одному из этих видов...» [13, с. 110] и сопровождать соответствующим комментарием. Если в этом разделе не указаны никакие действия, соответствующая СО является бессодержательной (не приводит к изменениям в системе), что является ошибкой.

Для описания системных операций следует придерживаться рекомендуемого шаблона с семью полями. Отступление от этой рекомендации является ошибкой.

В таблице 8.1 представлены типичные ошибки, допускаемые студентами при описании системных операций. Фрагменты описания системных операций, которые представлены в этой таблице, могут содержать более одной ошибки в соответствующей строке; рассматриваемая ошибка выделена серым фоном текста, а на изображениях – красным контуром.

Таблица 8.1 – Ошибки в описании системных операций

1	2
<p>8.1 Отсутствие соответствия имени системного сообщения в SD и имени системной операции</p>	<div style="text-align: center;">  </div> <p>Имя: изменитьСуммуТранзакции (Транзакция, суммаТранзакции) Обязанности: изменить сумму транзакции. Ссылки: прецедент «Редактирование транзакции».</p>
<p>Каждому системному сообщению должна соответствовать СО. Имена системных сообщений и соответствующих им операций должны совпадать</p>	
<p>8.2 Отсутствие соответствия списка параметров системного сообщения в SD и в описании системной операции</p>	<div style="text-align: center;">  </div> <p>Имя: отправитьЗаметку Обязанности: Отправитель инициирует отправку заметки другому сотруднику, заполнив соответствующие поля (заголовок, тело заметки, список адресатов). ...</p>
<p>Списки параметров системных сообщений и списки параметров соответствующих им операций должны совпадать</p>	
<p>8.3 Отсутствие соответствия имени атрибута понятия в диаграмме понятий и имени атрибута объекта в описании системной операции</p>	<div style="text-align: center;">  </div> <p>Имя: выполнитьПроверку(объявление) Обязанности: Проверить объявление на соответствие критериям. Ссылки: Прецедент “Изменение данных предприятия”. Примечание: Необходимы права для просмотра атрибута. Исключения: нет Предусловия: В системе выбрано объявление. Постусловия: Система вернула атрибут результатПроверки.</p>
<p>У понятия «Объявление» отсутствует атрибут результатПроверки. Имена атрибутов понятия в диаграмме понятий и имена атрибутов объекта в описании системной операции должны совпадать</p>	

Продолжение таблицы 8.1

1	2
8.4 Отсутствие полей в описании	<p><u>Имя</u>: предоставитьСписокДоговоров()</p> <p><u>Обязанности</u>: сформировать и предоставить список договоров, который содержит в себе все номера договоров.</p> <p><u>Постусловия</u>: создан объект <u>списокДог:Договор</u>.</p>
Указаны только три поля описания. Рекомендуется использовать семь полей	
8.5 Неверное наименование полей описания	<p><u>Имя</u>: отправитьЗаметку(заголовок, телоЗаметки, спАдресатов)</p> <p><u>Обязанности</u>: Отправитель инициирует отправку заметки другому сотруднику, заполнив соответствующие поля.</p> <p><u>Ссылки</u>: Прецедент «Отправка заметки»</p> <p>...</p> <p><u>Имя</u>: получитьТоварИзСписка(артикул).</p> <p><u>Обязанности</u>: найти товар по его артикулу.</p> <p><u>Ссылки</u>: прецедент «Изменение товара».</p> <p><u>Примечание</u>: нет.</p> <p><u>Исключение</u>: нет.</p> <p><u>Предусловие</u>: должен быть товар.</p> <p><u>Постусловие</u>: создан объект :Товар со значением «артикул» у одноименного атрибута</p>
Наименования полей описания системной операции должны быть подчеркнуты. Наименования полей «Исключение», «Предусловие», «Постусловие» следует указать во множественном числе	
8.6 Неверное поле «Предусловия»	<p><u>Имя</u>: просмотретьПустойСчет(номерСчета)</p> <p><u>Обязанности</u>: Предоставить просмотр информации о счете.</p> <p><u>Ссылки</u>: Прецедент «Работа со счетом администратором»</p> <p><u>Примечание</u>: Использовать самый быстрый способ доступа к БД.</p> <p><u>Исключения</u>: нет</p> <p><u>Предусловия</u>: Администратор вошел в режим работы со счетом.</p> <p>...</p>
Действие изменения режима работы ИС (уровень данных / уровень представления) не имеет отношения к объектам системы и является ошибкой	
8.7 Отсутствие описания изменений в поле «Постусловия»	<p><u>Имя</u>: просмотретьПустойСчет(номерСчета)</p> <p><u>Обязанности</u>: Предоставить просмотр информации о счете.</p> <p><u>Ссылки</u>: Прецедент «Работа со счетом администратором»</p> <p><u>Примечание</u>: Использовать самый быстрый способ доступа к БД.</p> <p><u>Исключения</u>: нет</p> <p><u>Предусловия</u>: нет</p> <p><u>Постусловия</u>: нет</p>
Если в результате выполнения системной операции в системе не происходит изменений, такая операция является бессодержательной, а ее присутствие в списке системных операций является ошибкой	
8.8 Неверное имя объекта в поле «Постусловия»	<p><u>Имя</u>: получитьСписокПустыхСчетов()</p> <p><u>Обязанности</u>: Предоставить просмотр списка счетов.</p> <p><u>Ссылки</u>: Прецедент «Работа со списком счетов»</p> <p><u>Примечание</u>: Использовать самый быстрый способ доступа к БД.</p> <p><u>Исключения</u>: нет</p> <p><u>Предусловия</u>: нет</p> <p><u>Постусловия</u>: Создан объект: <u>СписокПустыхСчетов</u> (создание экземпляра).</p>
Лексика имени объекта такая же, как в диаграмме последовательностей. Имя объекта должно быть подчеркнуто	

Продолжение таблицы 8.1

1	2
8.9 Неверное имя атрибута объекта в поле «Постусловия»	<p><u>Имя:</u> отправитьЗаметку (заголовок, телоЗаметки, спАдресатов) <u>Обязанности:</u> Отправитель отправляет заметку другому сотруднику. <u>Ссылки:</u> Прецедент «Отправка заметки» <u>Примечание:</u> Использовать самый быстрый способ доступа к БД. <u>Исключения:</u> нет <u>Предусловия:</u> Отправитель заполнил все поля и соблюдает условия, что он не может быть адресатом <u>Постусловия:</u> – Создан новый объект :Заметка (создание экземпляра) – Атрибут: Заметка.заголовок принял значение «заголовок» (модификация атрибута). ...</p>
Имя атрибута должно быть представлено именем существительным, начинаться с прописной буквы и не содержать символы, недопустимые в идентификаторах. Рекомендуется использование венгерской нотации Имя атрибута отделяется от имени объекта точкой. Имя атрибута должно быть подчеркнуто	
8.10 Отсутствие комментария в поле «Постусловия»	<p><u>Имя:</u> отправитьЗаметку (заголовок, телоЗаметки, спАдресатов) <u>Обязанности:</u> Отправитель отправляет заметку другому сотруднику. <u>Ссылки:</u> Прецедент «Отправка заметки» <u>Примечание:</u> Использовать самый быстрый способ доступа к БД. <u>Исключения:</u> нет <u>Предусловия:</u> Отправитель заполнил все поля и соблюдает условия, что он не может быть адресатом <u>Постусловия:</u> – создан новый объект :Заметка ...</p>
В поле «Постусловия» осуществляемые действия следует сопровождать комментариями. Текст комментария помогает обнаружить ошибки, указанные в 8.7 и 8.13	
8.11 Неверный комментарий в поле «Постусловия»	<p><u>Имя:</u> получитьСписокПустыхСчетов () <u>Обязанности:</u> Предоставить возможность просмотреть список счетов. <u>Ссылки:</u> Прецедент «Работа со списком счетов» <u>Примечание:</u> Использовать самый быстрый способ доступа к БД. <u>Исключения:</u> нет <u>Предусловия:</u> нет <u>Постусловия:</u> Создан объект :СписокПустыхСчетов {создание экземпляра}.</p>
Текст комментария следует заключать в круглые скобки	
8.12 Отсутствие значений для выполнения системной операции	<p><u>Имя:</u> изменитьПолеНазваниеЗадания() <u>Обязанности:</u> предоставить возможность изменить название задания. <u>Ссылки:</u> Прецедент «Редактирование задания» <u>Примечание:</u> использовать самый быстрый способ доступа к БД. <u>Исключения:</u> нет. <u>Предусловия:</u> Администратор вошел в режим работы редактирования задания. <u>Постусловия:</u> атрибут :Задание.названиеЗадания принял значение «название» (модификация атрибута).</p>
В поле «Постусловия» использованы имена объектов, атрибутов объектов и литералы, которых не определены / не заданы в этой операции	

Продолжение таблицы 8.1

1	2
8.13 Бессодержательная системная операция	<p><u>Имя</u>: получитьТоварИзСписка(артикул). <u>Обязанности</u>: найти товар по его артикулу. <u>Ссылки</u>: прецедент «Изменение товара». <u>Примечание</u>: нет. <u>Исключение</u>: нет. <u>Предусловие</u>: должен быть товар. <u>Постусловие</u>: возвращает объект :Товар</p> <p><u>Имя</u>: подтвердитьИзмененияТранзакции(транз) <u>Обязанности</u>: сохранить измененную транзакцию в системе. <u>Ссылки</u>: Прецедент “Работа с транзакцией” <u>Примечание</u>: Необходимы права для изменения атрибута. <u>Исключения</u>: нет <u>Предусловия</u>: Были изменены атрибуты объекта транз:Транзакции. <u>Постусловия</u>: Объект транз:Транзакции сохраняется в системе</p>
<p>В поле «Постусловия» следует описывать те действия, которые приводят к одному из трех видов изменений в системе на уровне данных. Действия «возвращает» и «сохраняется» не соответствуют ни одному из этих видов изменений, поэтому системные операции являются бессодержательными, а их присутствие в списке системных операций – ошибка. Обнаружить эту ошибку можно было с помощью указания комментария, но он в поле «Постусловие» отсутствует</p>	
8.14 Лишние действия в поле «Постусловия»	<p><u>Имя</u>: получитьСписокПустыхСчетов () <u>Обязанности</u>: Предоставить возможность просмотреть список счетов. <u>Ссылки</u>: Прецедент «Работа со списком счетов» <u>Примечание</u>: Использовать самый быстрый способ доступа к БД. <u>Исключения</u>: нет <u>Предусловия</u>: нет <u>Постусловия</u>: Для новой работы создан объект :СпПустыхСчетов (создание экземпляра).</p> <p><u>Имя</u>: добавитьПредприятие (предприятие) <u>Обязанности</u>: Предоставить возможность добавить новое предприятие. <u>Ссылки</u>:Прецедент «Работа администратора с предприятиями» <u>Примечание</u>: Предполагает также подтверждение объявления. <u>Исключения</u>: нет. <u>Предусловия</u>: Администратор получил список объявлений. <u>Постусловия</u>: Для новой работы создан объект :Предприятие (создание экземпляра).</p>
<p>Действия «Для нового (-ой)...» следует указывать только в случае, если системной операцией: а) создается объект, который является элементом контейнера, б) создается этот контейнер, в) созданный объект добавляется в новый контейнер. При этом действия «Для нового (-ой)...» следует отнести к контейнеру, а не к его элементу.</p>	

Продолжение таблицы 8.1

1	2
8.15 Отсутствие в поле «Постусловия» действия, предусмотренного системной операцией	<p><u>Имя:</u> удалитьОбъявление (названиеОбъявления) <u>Обязанности:</u> Предоставить возможность удалить объявление. <u>Ссылки:</u> Прецедент «Работа администратора с объявлениями». <u>Примечание:</u> Использовать самый быстрый способ доступа к БД. <u>Исключения:</u> нет. <u>Предусловия:</u> нет. <u>Постусловия:</u> нет.</p>
<p>Эта операция должна привести к изменению на уровне данных в поле «Постусловия»: – разрыв связи (между объектом класса Объявление, у которого соответствующий атрибут имеет значение «названиеОбъявления», и контейнером, моделирующим список объявлений), – уничтожение экземпляра (указанного объекта), если между списком объявлений и его элементами установлена композитная агрегация. Подобная ошибка допускается и для операции с именем «добавить...»</p>	
8.16 Отсутствие использования параметра системной операции для ее выполнения	<p><u>Имя:</u> выбратьКонкретнуюТранзакцию(номерСчета, номерТранзакции) <u>Обязанности:</u> Предоставить выбор транзакции из списка. <u>Ссылки:</u> Прецедент «Работа со списком транзакций» <u>Примечание:</u> Использовать самый быстрый способ доступа к БД. <u>Исключения:</u> нет. <u>Предусловия:</u> Кассир получил список транзакций. <u>Постусловия:</u> нет.</p>
	<p><u>Имя:</u> изменитьСтоимостьТовара(товар, цена). <u>Обязанности:</u> изменить стоимость товара. <u>Ссылки:</u> прецедент «ИзменениеТовара». <u>Примечание:</u> нет <u>Исключения:</u> нет <u>Предусловия:</u> нет <u>Постусловия:</u> атрибут :ЦенаТовара принял значение «цена» (модификация атрибута).</p>
<p>Параметры системной операцией передают данные, которые следует использовать в поле «Постусловия» для изменений в системе. Если все параметры системной операции «лишние», то это свидетельствуют о бессодержательности операции</p>	
8.17 Подмена изменения объекта изменением контейнера	<div data-bbox="555 1464 1305 1621" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> </div> <p><u>Имя:</u> удалитьОбъявление(объявление) <u>Обязанности:</u> Удалить объявление из системы. <u>Ссылки:</u> Прецедент “Изменение данных предприятия”. <u>Примечание:</u> Необходимы определенные права для удаления объявления. <u>Исключения:</u> Нет. <u>Предусловия:</u> В системе выбрано объявление. <u>Постусловия:</u> :Объявление было удалено системой.</p>
<p>Имя «удалить...» указывает на то, что СО приводит к изменению на уровне данных для объектов-контейнеров и их элементов: «разрыв связи» и, возможно, «уничтожение экземпляра» в случае композиции. Система не является контейнером для объявлений. Поэтому системной операции следует дать имя «уничтожитьОбъявление»</p>	

Продолжение таблицы 8.1

1	2
8.18 Неверное имя класса объекта	<p><u>Имя:</u> выбратьКонкретныйПустойСчет(номерСчета) <u>Обязанности:</u> Предоставить возможность выбрать счет из списка. <u>Ссылки:</u> Прецедент «Работа со списком счетов» <u>Примечание:</u> Использовать самый быстрый способ доступа к БД. <u>Исключения:</u> нет. <u>Предусловия:</u> Администратор получил список пустых счетов. <u>Постусловия:</u> создан объект :списокПустыхСчетов (создание экземпляра). ... а)</p>
	<p><u>Имя:</u> подтвердитьИзмененияТранзакции(транз) <u>Обязанности:</u> сохранить измененную транзакцию в системе. <u>Ссылки:</u> Прецедент “Работа с транзакцией” <u>Примечание:</u> Необходимы права для изменения атрибута. <u>Исключения:</u> нет <u>Предусловия:</u> Были изменены атрибуты объекта транз:Транзакции. <u>Постусловия:</u> Объект транз:Транзакция сохраняется в системе б)</p>
	<p><u>Имя:</u> удалитьОбъявление(объявление) <u>Обязанности:</u> Удалить объявление из системы. <u>Ссылки:</u> Прецедент “Изменение данных предприятия”. <u>Примечание:</u> Необходимы определенные права для удаления объявления. <u>Исключения:</u> Нет. <u>Предусловия:</u> В системе выбрано объявление. <u>Постусловия:</u> :объявление было удалено системой. в)</p>
<p>Имя класса указано неверно: а) если «списокПустыхСчетов» – это имя класса-контейнера, то оно должно начинаться с заглавной буквы, б) в полях «Предусловия» и «Постусловия» используется один и тот же объект, но имена классов объекта в этих полях отличаются, в) использование одного имени для параметра системной операции и имени класса объекта недопустимо: параметром передаются объекты и/или данные, но не имя класса Также в ошибке 8.18, а) если объект класса «СписокПустыхСчетов» используется в поле «Постусловия» для описания других изменений в системе, то он не может быть анонимным</p>	

9. Создание диаграмм сотрудничества

Диаграммы сотрудничества имеют High LoD, т.к. представляют модель взаимодействия элементов системы на этапе проектирования. Но используя иерархическую нумерацию, «...в последовательности сообщений, образующей взаимодействие, можно показывать больше или меньше элементов...» [14].

В теме 9 выполняется переход от этапа анализа к этапу проектирования ИС. Поэтому важной является согласованность диаграмм этих этапов.

Диаграмма сотрудничества для соответствующего системного сообщения с точки зрения кодирования на языке объектно-ориентированного программирования представляет собой тело функции в виде вызовов других функций, реализующих изменение состояния объектов (составных частей) системы.

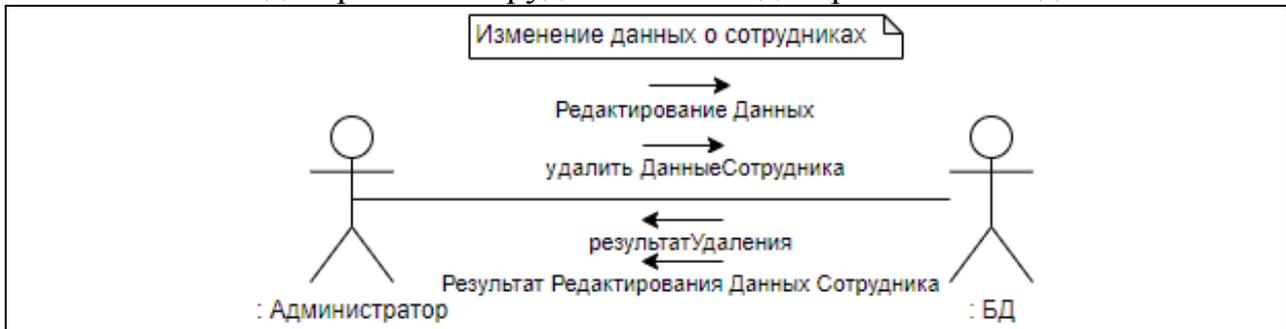
Поэтому должно поддерживаться полное соответствие:

- имени системного сообщения в SD, имени системной операции и имени системного сообщения в CollD,
- списка параметров (количества параметров и порядка их следования) у системного сообщения в SD, списка параметров системной операции и списка параметров системного сообщения в CollD.

Нарушение это соответствия является ошибкой.

Для создания диаграмм сотрудничества рекомендуется использовать сервис draw.io [15]. В отличие от GenMyModel с его помощью нельзя получить модель, возможно лишь создание графа диаграммы.

На рисунке 9.1 представлена диаграмма, которая иллюстрирует непонимание отличия диаграммы сотрудничества от диаграммы последовательностей.



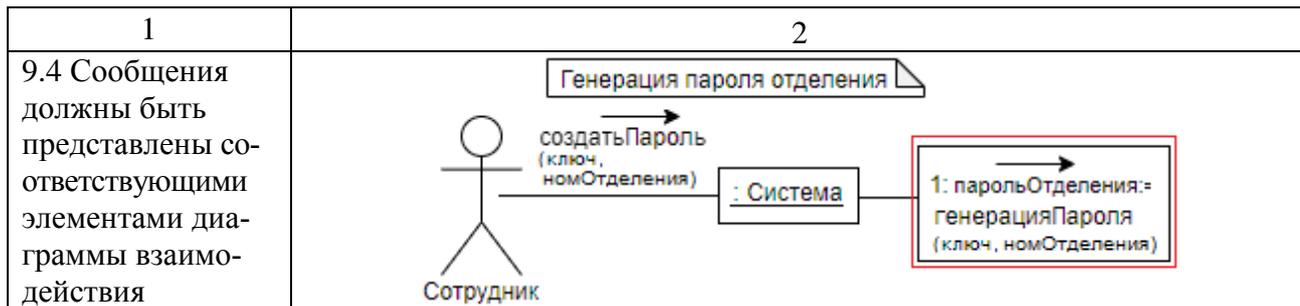
На рисунке 9.1 – Модель со смешением понятий CollD и SD

В таблице 9.1 представлены типичные ошибки, допускаемые студентами при создании CollD. Фрагменты диаграмм, которые представлены в этой таблице, могут содержать более одной ошибки в соответствующей строке; рассматриваемая ошибка выделена красным контуром.

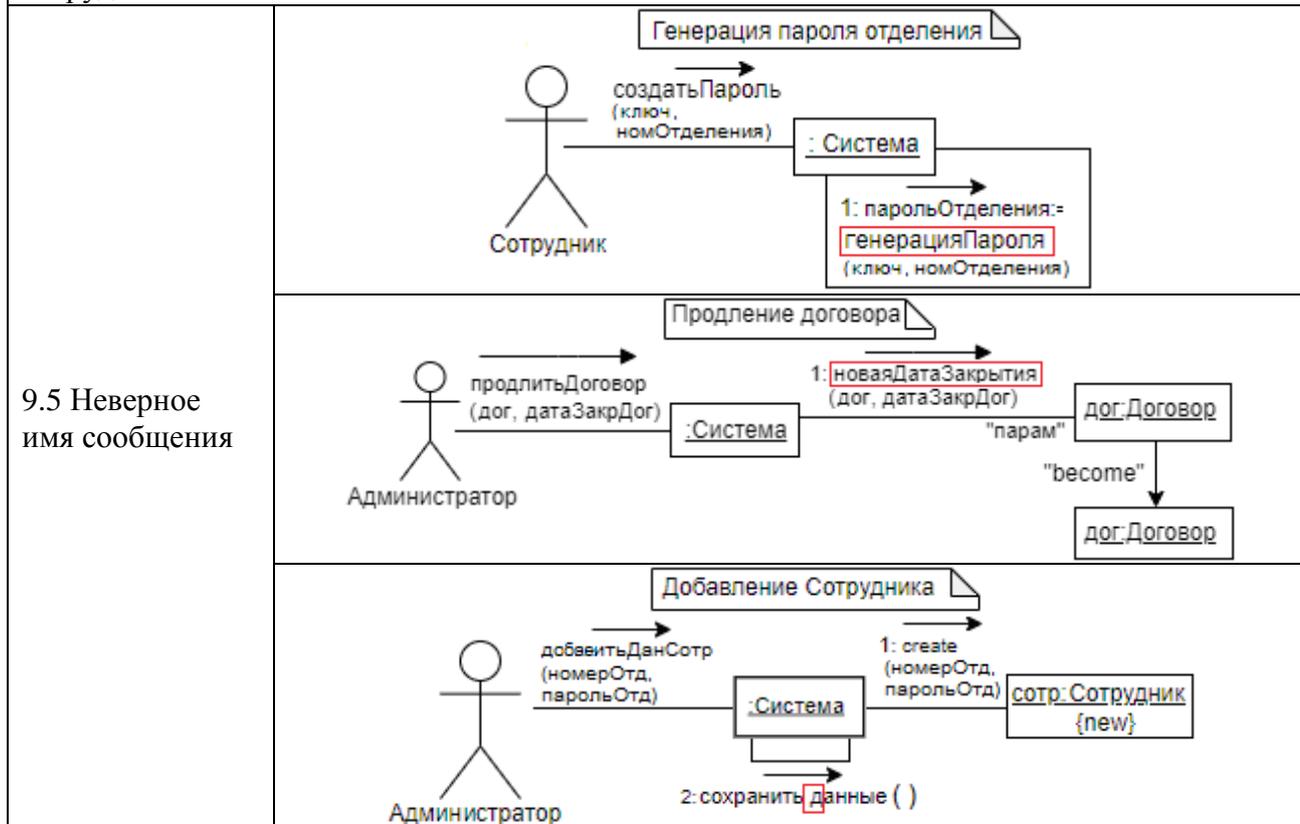
Таблица 9.1 – Ошибки в диаграммах сотрудничества

Номер и текст ошибки	Пример с ошибкой
1	2
Лексические ошибки	
9.1 Неверное имя системы	<p style="text-align: center;">Получить счет из списка</p>
Система является объектом, поэтому ее имя должно быть подчеркнуто	
9.2 Неверное имя актора	<p style="text-align: center;">Получить детали транзакции</p>
	<p style="text-align: center;">Получить детали транзакции</p>
Лексика имени актора не изменяется, она единая на всех этапах и шагах объектно-ориентированного анализа, проектирования и реализации	
9.3 Неверное имя объекта	<p style="text-align: center;">Получить детали транзакции</p>
	<p style="text-align: center;">Получить счет из списка</p>
Лексика имени объекта такая же, как в диаграмме последовательностей	

Продолжение таблицы 9.1



Прямоугольниками в диаграмме сотрудничества обозначаются объекты или классы. Для отображения сообщений используются стрелки вдоль линий ассоциаций, представляющих сотрудничество объектов

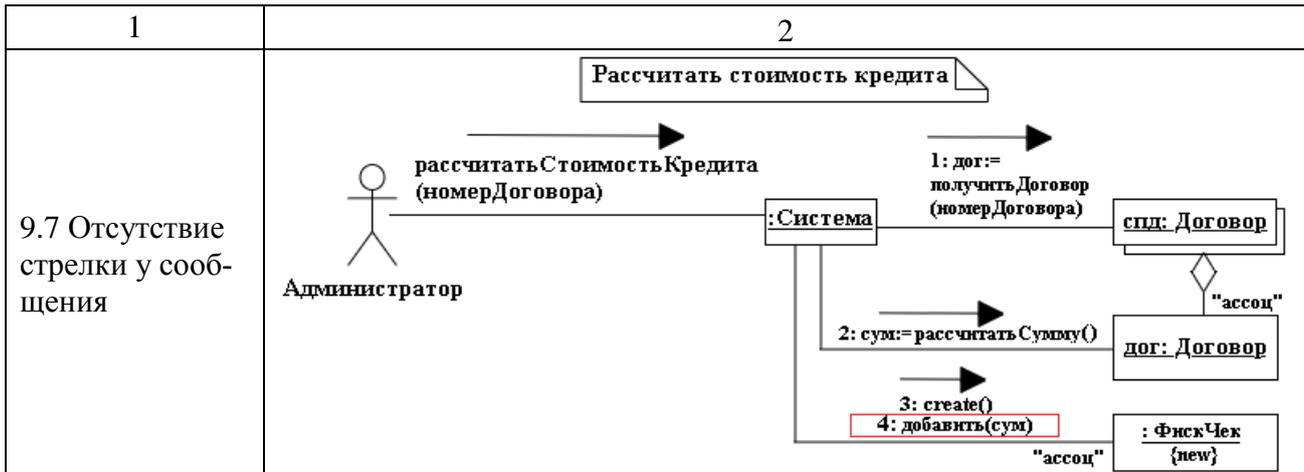


Лексика имени сообщения такая же, как в диаграмме последовательностей. Имена системных сообщений в диаграмме последовательностей и диаграмме сотрудничества должны совпадать с именем системной операции



Имя параметра сообщения должно быть представлено именем существительным, начинаться с прописной буквы и не содержать символы, недопустимые в идентификаторах. Рекомендуется использование венгерской нотации

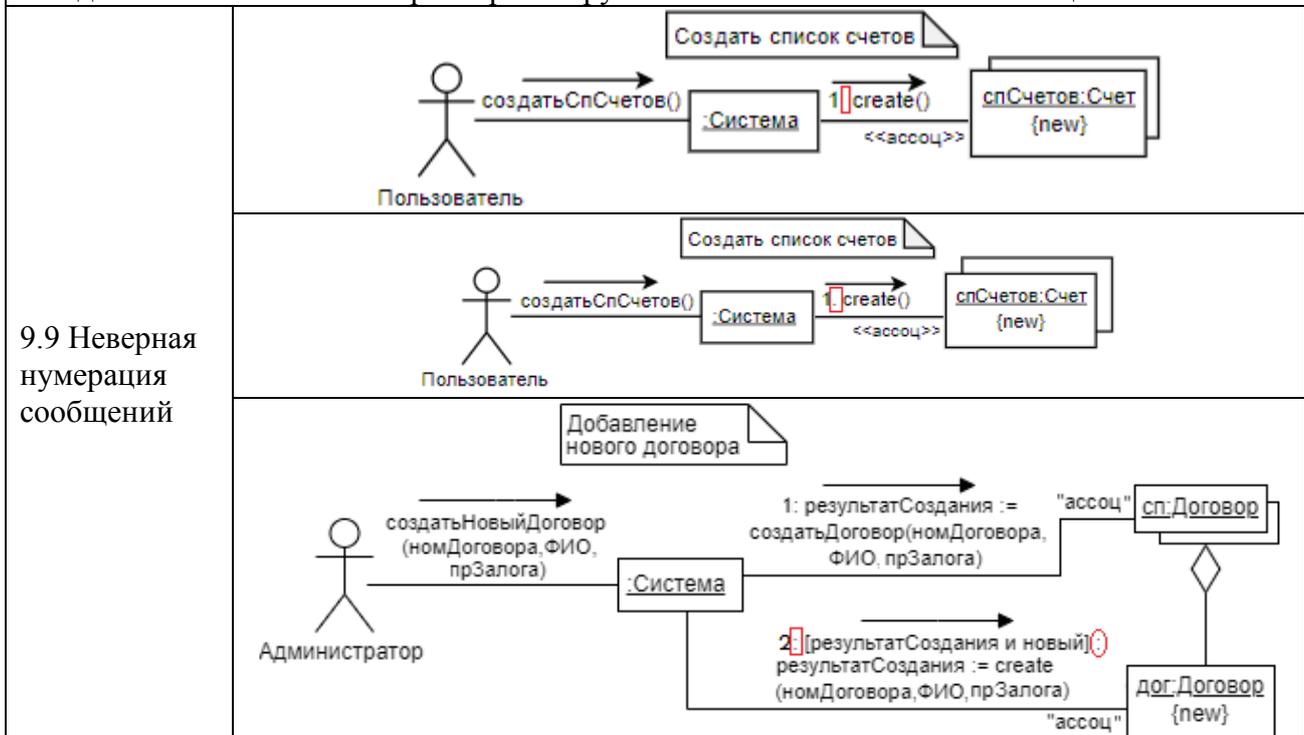
Продолжение таблицы 9.1



Вдоль линии сотрудничества может быть установлено взаимодействие между объектами путем передачи одного или нескольких сообщений. Каждое сообщение должно иметь метку и сопровождающую ее стрелку, которая иллюстрирует направление передачи сообщения



Диаграммы сотрудничества имеют высокий уровень детализации, поэтому метка сообщения должна иметь список параметров в круглых скобках после имени сообщения

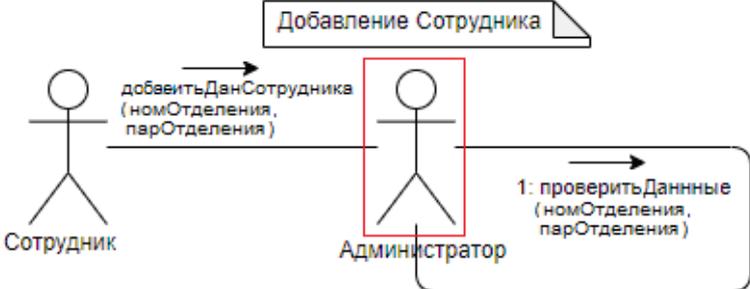
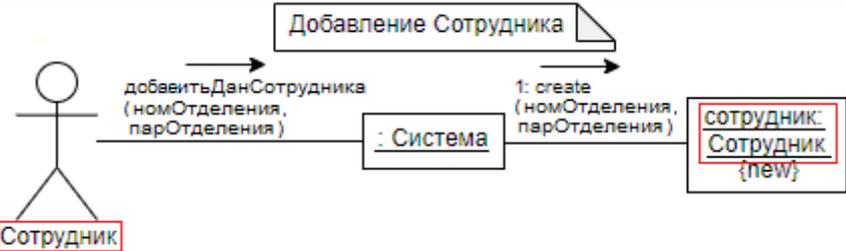
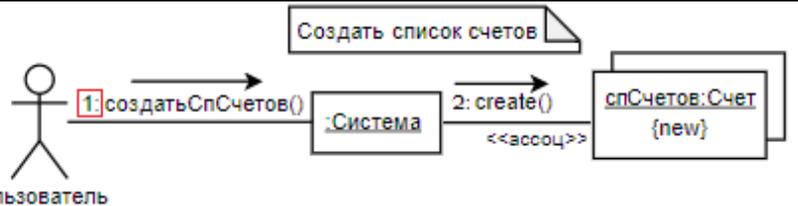


Номер сообщения должен быть отделен от следующих за ним частей метки сообщения символом двоеточия. Указание двоеточия перед условием отправки сообщения является ошибкой

Продолжение таблицы 9.1

1	2
<p>9.10 Неверное обозначение мультиобъекта</p>	
<p>Графический символ мультиобъекта образуется повторением символа объекта (прямоугольника). Использованный в этой модели символ для обозначения мультиобъекта не является лексемой диаграммы сотрудничества</p>	
<p>9.11 Неверное обозначение ключевого слова для описания времени жизни объекта</p>	
<p>Ключевое слово для описания времени жизни объектов не следует подчеркивать</p>	
<p>9.12 Неверное обозначение стереотипа полюса роли ассоциации</p>	
<p>Стереотип полюса роли ассоциации следует заключать в парные кавычки</p>	
<p>9.13 Неверное обозначение стереотипа зависимости</p>	
<p>Стереотип зависимости следует заключать в парные кавычки</p>	

Продолжение таблицы 9.1

1	2
Синтаксические ошибки	
<p>9.14 Отсутствие системы</p>	
<p>Диаграмма сотрудничества является моделью этапа проектирования, на котором система рассматривается как «белый ящик». Наличие такого элемента, как система, в этом виде диаграмм является обязательным. Актор не может быть использован для моделирования системы, т.к. он представляет внешнюю среду и является источником сообщений, поступающих в систему</p>	
<p>9.15 Актор одновременно за границей системы и внутри нее</p>	
<p>Невозможно использовать в модели одну и ту же сущность в роли внешней среды (актор всегда находится за границей системы) и как часть этой же системы</p>	
<p>9.16 Имя входного (системного) сообщения и сообщений между объектами системы не должны совпадать</p>	
<p>Нарушение принципа идентификации. Ошибка аналогична ошибке 7.19</p>	
<p>9.17 Лишняя нумерация сообщения</p>	
<p>В диаграммах сотрудничества нумерация используется для моделирования времени при отсутствии оси времени: «...номер сообщения определяет его положение в цепочке сообщений...» [11, с. 117]. Системное (входящее) сообщение не требует нумерации</p>	

Продолжение таблицы 9.1

1	2
<p>9.18 Лишняя нумерация за- висимости</p>	
<p>Стереотип зависимости не является сообщением, поэтому не требует нумерации</p>	
<p>9.19 Неверное размещение стереотипа за- висимости</p>	
<p>Зависимость со стереотипом «become» проводится от объекта в исходном состоянии к тому же самому объекту, но в измененном состоянии, имя которого указывают в квадратных скобках</p>	
<p>9.20 Несоответствие типов данных у мультиобъекта и его элемента</p>	
<p>Элементы мультиобъекта имеют тип данных, который указывается в имени мультиобъекта после двоеточия. Необходимо контролировать совпадение типов данных мультиобъекта и его элемента, связанных агрегацией / композицией. В этой модели объект «спТоваров» должен иметь тип данных Товар</p>	
<p>9.21 Отсутствие агрегации / композиции между объекта- ми</p>	
<p>Между объектом типа «коллекция» / «мультиобъект» и объектом типа «элемент коллекции» / «элемент мультиобъекта» следует указать отношение агрегации / композиции</p>	

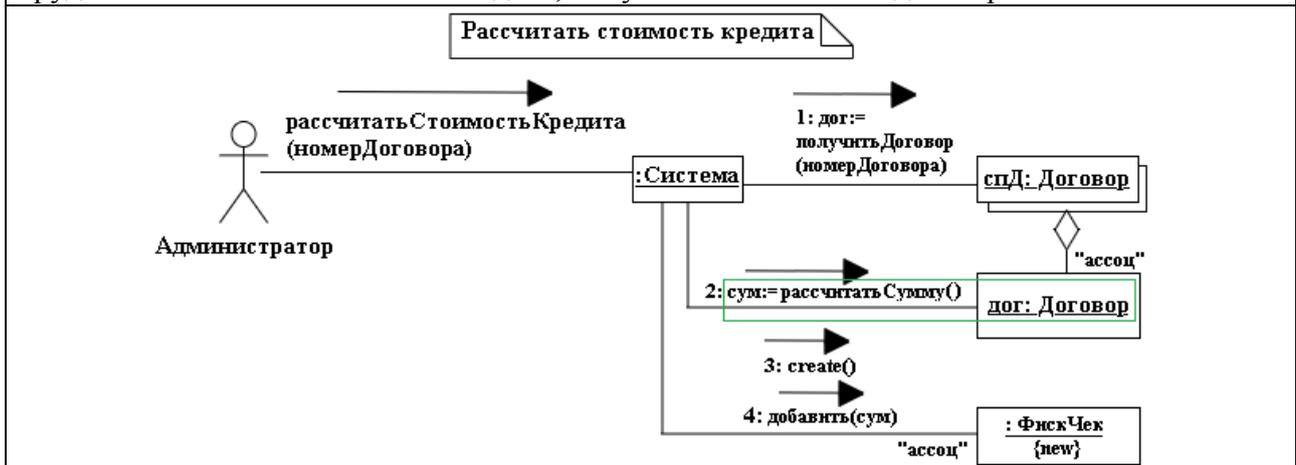
Продолжение таблицы 9.1

1	2
Семантические ошибки	
<p>9.22 Недопустимое отношение обобщения</p>	
<p>В диаграммах сотрудничества отношение обобщения не допускается. Символ обобщения также не является лексемой этого вида диаграмм</p>	
<p>9.23 Неверное имя сообщения</p>	
<p>Сообщение с именем «добавить» должно соответствовать выполнению в системе операции добавления элемента в коллекцию / мультиобъект. Адрес магазина является атрибутом объекта класса «ЗаголовокЧека». Добавлять динамически новые атрибуты класса невозможно. Поэтому в модели имя сообщения «добавить» следует заменить на имя, соответствующее операции установки значения атрибута, которое передано параметром</p>	
<p>9.24 Бессодержательное сообщение</p>	
<p>Сохранение состояния объекта не приводит к изменениям в объектно-ориентированной системе, поэтому любые сообщения о сохранении являются бессодержательными. Ошибка, связанная с действием сохранения (уровень данных), рассмотрена в ошибке 4.40, жс</p>	

Продолжение таблицы 9.1



Объект «спД» (список договоров) не имеет атрибутов для расчета суммы кредита для конкретного договора: дату открытия договора, дату закрытия договора, стоимость предмета залога. Поэтому поручить выполнение такой операции системе этот список не может. Необходимые атрибуты для расчета суммы кредита имеет объект «дог» класса Договор. Поэтому сообщение «рассчитатьСумму» должно быть отправлено системой вдоль линии сотрудничества системы и объекта «дог», полученного из списка договоров:



Сообщение с именем «установитьДату» отправлено вдоль линии связи между классом «Системная дата» и объектом архЗаметка. Подобный класс в языках программирования может предоставить своим клиентам значение текущей даты, но не установить это значение в каком-то объекте

Продолжение таблицы 9.1

1	2
<p>9.27 Отсутствие сообщения классу о текущей дате и времени</p>	<p>Продление договора</p>
<p>Текущая дата и время должны быть получены от соответствующего класса до их использования. Сообщение такому классу возвращает значение системного времени</p>	<p>Прочитать заметку</p>
<p>9.28 Лишний параметр сообщения</p>	<p>Редактирование договора</p>
<p>9.29 Неверное имя параметра сообщения в системе</p>	<p>Чтение заметки</p>
<p>Имя объекта, созданного сообщением с номером 1, и параметра сообщения с номером 2 не совпадают, поэтому добавление нового объекта в мультиобъект невозможно</p>	

Продолжение таблицы 9.1

1	2
<p>9.30 Неверное / лишнее ответное сообщение</p>	
<p>Ответные сообщения используются при обмене асинхронными сообщениями. Обе модели имеют синхронные сообщения, для которых ответные сообщения не требуются</p>	
<p>9.31 Отсутствие объекта как результата сообщения</p>	
<p>Отсутствие объекта (элемента мультиобъекта) как результата сообщения с номером 1. Также должна присутствовать агрегация / композиция между мультиобъектом и этим объектом</p>	
<p>9.32 Неверный результат сообщения о создании объекта и его использование</p>	
<p>Два сообщения (с номерами 1 и 2) имеют одинаковые списки параметров и одинаковое имя переменной результата. Создание объекта в мультиобъекте (сообщение с номером 1) невозможно. Объект с именем «дог» должен быть создан конструктором (сообщением с номером 2 без указания имени переменной результата), а затем добавлен в мультиобъект с именем «спД»</p>	
<p>9.33 Отсутствие стереотипа полюса роли ассоциации</p>	
<p>Стереотип полюса роли ассоциации следует указывать с учетом их приоритетов</p>	

Продолжение таблицы 9.1

1	2
<p>9.34 Ключевое слово для описания времени жизни объекта не соответствует сообщению</p>	
<p>Ключевое слово для описания времени жизни объекта как созданного не соответствует сообщению с именем «получить». Имя объекта «заявка» использовано в двух статусах: как входного параметра системного сообщения и как имя нового объекта. Указание ключевого слово «new» является ошибкой, т.к. не соответствует времени жизни этого объекта. Если необходимо было найти какую-то заявку, то параметром (-ами) системного сообщения следует указать имя (имена) атрибута (-ов) заявки для ее поиска</p>	
<p>9.35 Лишнее ключевое слово для описания времени жизни объекта</p>	
<p>Указание ключевого слова «new» для описания времени жизни мультиобъектов является ошибкой, т.к. соответствующие списки будут создаваться каждый раз при создании одного элемента такого мультиобъекта</p>	
<p>9.36 Отсутствие идентификации изменения состояния объекта</p>	
<p>Отсутствует указание начального и конечного состояния объекта, к которому применялась операция редактирования, и стереотипа зависимости «become»</p>	

Продолжение таблицы 9.1

1	2
<p>9.37 Отсутствие идентификации изменения состояния мультиобъекта / коллекции после добавления элемента</p>	<p>Создание заявки</p> <p>ВладелецПредприятия → Система: создатьЗаявку(email, владелецПредприятия)</p> <p>Система → заявка:Заявка {new}: 1: create(email, владелецПредприятия) <<ассоц>></p> <p>Система → списокЗаявок:Заявка: 2: добавить(заявка) <<парам>></p> <p>заявка:Заявка {new} *-- списокЗаявок:Заявка</p>
<p>Отсутствует указание начального и конечного состояния мультиобъекта, к которому применялась операция добавления элемента, и стереотип зависимости «become»</p>	
<p>9.38 Отсутствие идентификации изменения состояния мультиобъекта / коллекции после удаления элемента</p>	<p>Прочитать заметку</p> <p>Получатель → Система: прочитатьЗаметку(непрЗаметка)</p> <p>Система → архивнЗам: АрхЗам {new}: 1: create(непрЗам) <<ассоц>></p> <p>Система → спНЗам: НЗам: 2/3: удалить(непрЗаметка) <<ассоц>></p> <p>Система → спАрхЗам: АрхЗам: 1/2: добавить(архивнЗам) <<парам>></p> <p>архивнЗам: АрхЗам {new} *-- спНЗам: НЗам</p> <p>спАрхЗам: АрхЗам *-- архивнЗам: АрхЗам {new}</p>
<p>Отсутствует указание начального и конечного состояния мультиобъекта, к которому применялась операция удаления элемента, и стереотип зависимости «become»</p>	
<p>9.39 Стереотип полюса роли ассоциации использован для зависимости</p>	<p>Редактирование договора</p> <p>Администратор → Система: изменитьДоговор(номДоговора, ФИО)</p> <p>Система → спД: Договор: 1: дог := получитьДоговор(номДоговора) "парам"</p> <p>Система → дог: Договор[исх]: 1/2: изменитьФИОКлиента(ФИО) "парам"</p> <p>спД: Договор *-- дог: Договор[исх]</p> <p>дог: Договор[исх] *-- дог: Договор[изм] "ассоц" "become"</p>
<p>Стрелка зависимости не является обозначением сотрудничества двух объектов, поэтому вдоль нее сообщение не может быть передано. У зависимости указывать стереотип полюса роли ассоциации не требуется</p>	

10. Создание CD

Диаграммы классов как модели этапа проектирования являются диаграммами с высоким уровнем детализации [7, 8].

В них используются синтаксические функции для повышения уровня детализации. Это такие субъекты, как атрибуты с указанием их типов данных, операции с указанием их списка параметров и типов данных каждого параметра и результата операции, имена ассоциаций и их роли, направления навигации, зависимости, область видимости частей класса.

Все элементы диаграммы понятий и диаграммы сотрудничества, выделенные в темах 6 и 9, должны быть представлены на диаграмме классов. Поэтому необходимо соблюдать выполнение тех же требований к именам и набору отношений, которые были сформулированы для концептуального моделирования. Требования к соответствию диаграммы классов и диаграммы сотрудничества сформулированы в [13, с. 126].

Перед созданием диаграммы классов необходимо выбрать целевой язык ООП, который определяет набор типов данных, принятый к использованию в CD, и набор встроенных контейнеров для моделирования агрегации / композиции.

Лексическими являются ошибки, которые нарушают правила создания и использования лексем подмножества UML для создания CD – допустимых графических образов классов, интерфейсов, перечислений и перечисленных выше субъектов и их именованя.

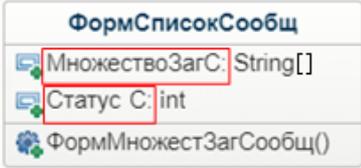
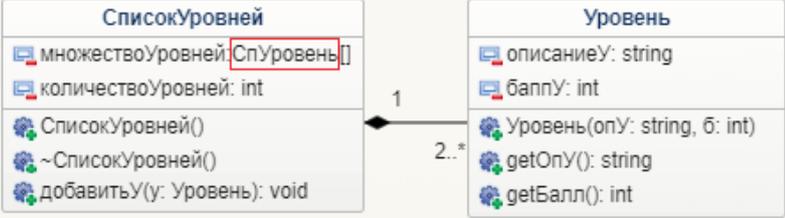
Синтаксические ошибки возникают при отсутствии классов, интерфейсов и перечислений / субъектов, неверном установлении связей между классами, интерфейсами и перечислениями и/или их размещении по отношению друг к другу, а также при нарушении принципа идентификации.

К семантическим ошибкам приводит использование лишних атрибутов и ассоциированных с ними операций, неверная спецификация типов данных атрибутов, параметров и результатов операций, неверное понимание моделирования отношений между классами, интерфейсами и перечислениями.

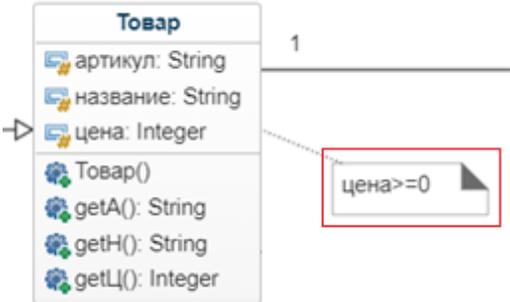
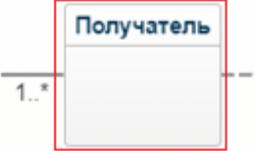
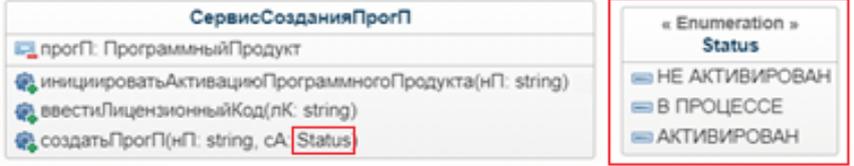
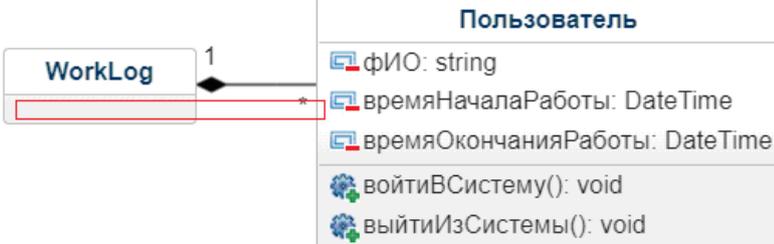
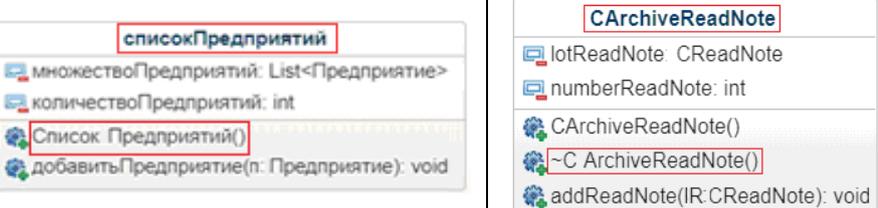
Для создания диаграмм классов следует использовать сервис genMyModel.

В таблице 10.1 представлены типичные ошибки, допускаемые студентами при создании диаграмм классов с High LoD. Фрагменты диаграмм, которые представлены в этой таблице, могут содержать более одной ошибки в соответствующей строке; рассматриваемая ошибка выделена красным контуром.

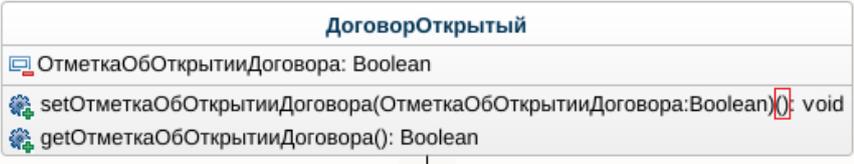
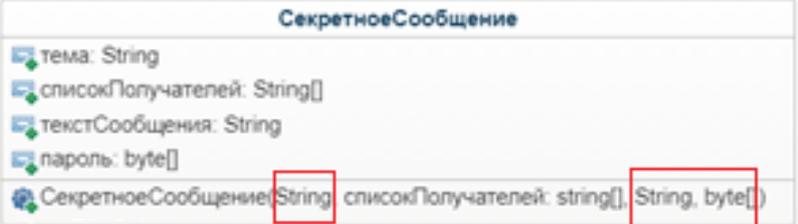
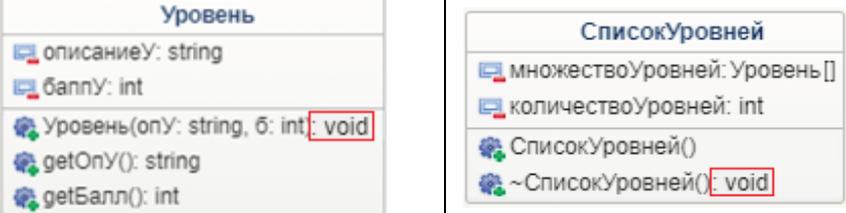
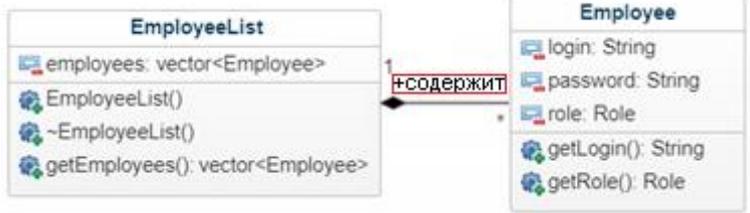
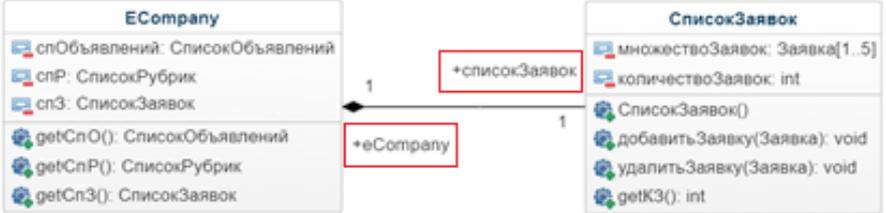
Таблица 10.1 – Ошибки в диаграммах классов с высоким уровнем детализации

Номер и текст ошибки	Пример с ошибкой
1	2
Лексические ошибки	
<p>10.1 Использование элементов, которые не являются лексемами для диаграммы классов</p>	
Элемент «Актор» не входит в список лексем, разрешенный стандартом UML для CD	
<p>10.2 Неверное имя класса / интерфейса / перечисления</p>	
<p>Имена классов, интерфейсов, перечислений не должны выражать действие. Эти имена должны быть представлены именем существительным, начинаться с заглавной буквы и не содержать символы, недопустимые в идентификаторах</p>	
<p>10.3 Неверное имя атрибута класса</p>	
<p>Имя атрибута класса должно быть представлено именем существительным, начинаться с прописной буквы и не содержать символы, недопустимые в идентификаторах. Рекомендуется использование венгерской нотации. Имя атрибута класса не должно выражать действие</p>	
<p>10.4 Неверное имя операции класса</p>	
<p>Имя операции класса должно быть представлено глаголом в повелительном наклонении, начинаться с прописной буквы (кроме конструкторов и деструкторов) и не содержать символы, недопустимые в идентификаторах. Рекомендуется использование венгерской нотации. Имя операции класса должно выражать действие</p>	
<p>10.5 Неверное имя типа данных атрибута, аргумента операции, результата операции</p>	
<p>Имена типов данных должны соответствовать именам, используемым в целевом языке ООП, или именам других классов модели</p>	

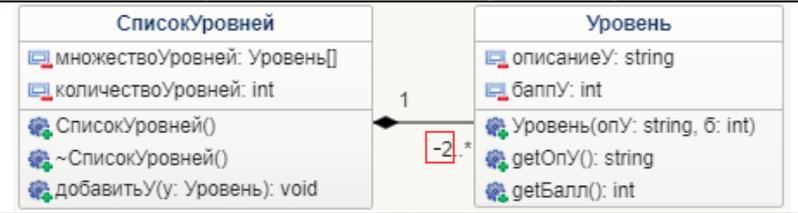
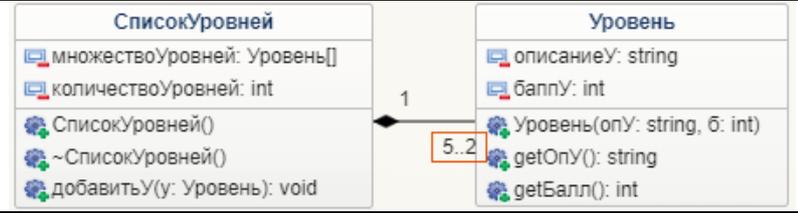
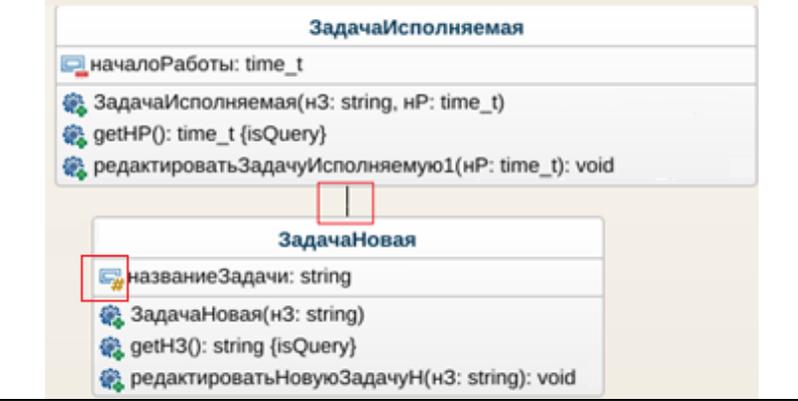
Продолжение таблицы 10.1

1	2
10.6 Неверное указание ограничения	
Текст ограничения должен быть заключен в фигурные скобки	
Синтаксические ошибки	
10.7 Класс или интерфейс не содержит ни атрибутов, ни операций	
Классы (кроме участвующих в обобщении) и интерфейсы должны содержать хотя бы один атрибут или операцию	
10.8 Существует класс, интерфейс или перечисление, не имеющее никаких ассоциаций	
У каждого класса, интерфейса или перечисления должна существовать хотя бы одна ассоциация с другими классами. Без связей такие элементы диаграммы могут выполнять только роль пассивного хранилища данных	
10.9 Отсутствие атрибута для реализации агрегации / композиции	
Класс-целое должен обязательно иметь атрибут для объекта класса-части	
10.10 Отсутствие совпадения имени конструктора / деструктора с именем класса	
Конструктор и деструктор класса отвечают за создание и уничтожение экземпляров этого класса. Поэтому концепцией объектно-ориентированного проектирования и программирования определено совпадение имен этих операций с именем соответствующего класса	

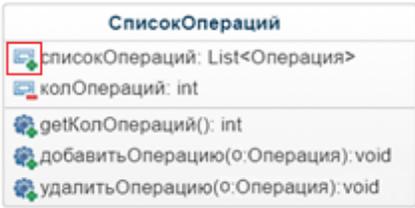
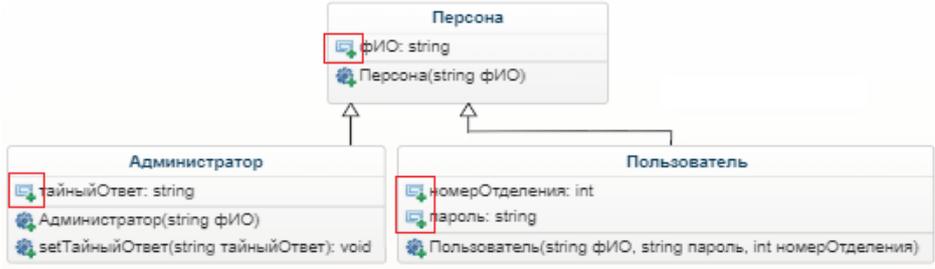
Продолжение таблицы 10.1

1	2
10.11 Лишние символы в описании операции	
Список параметров операции заключается в круглые скобки, их повторение является ошибкой	
10.12 Неверный список параметров операции	
В списке параметров операции для каждого параметра следует указывать его имя и тип данных, разделенные двоеточием	
10.13 Отсутствие имени аргумента в описании операции	
В описании операции для каждого ее аргумента следует указать его имя и имя типа данных	
10.14 Указание типа результата у конструктора / деструктора	
Концепцией объектно-ориентированного проектирования и программирования определено, что конструктор и деструктор не могут иметь тип результата для предотвращения доступа к объекту способом, отличным от доступа по его имени	
10.15 Лишнее имя у символа агрегации / композиции	
Для именования агрегации / композиции использовано имя роли. Отношения агрегации и композиции изображаются соответствующими лексемами диаграммы, поэтому их именование не требуется	
10.16 Избыточное имя роли агрегации / композиции	
Отношения агрегации и композиции изображаются соответствующими лексемами диаграммы, поэтому именование их ролей не требуется	

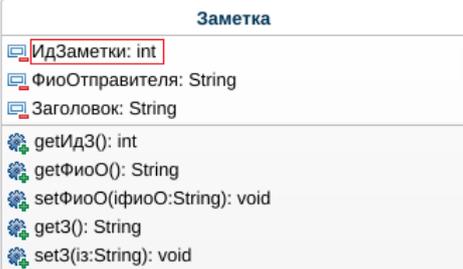
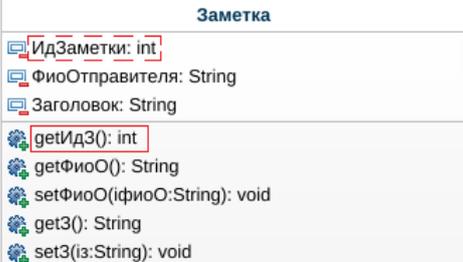
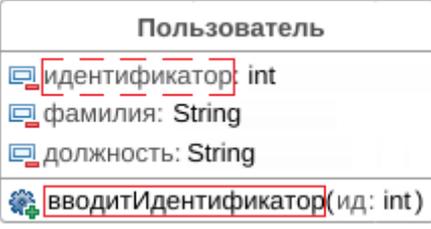
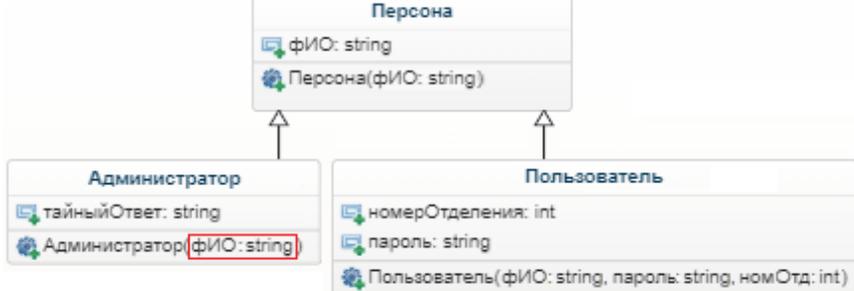
Продолжение таблицы 10.1

1	2
<p>10.17 Отсутствие кратности у роли агрегации / композиции</p>	 <pre> classDiagram class Project { name: string Project(n: string) Project() getName(): string setName(n: string): bool } class ProjectService { projects: vector<Project> add(p: Project): bool remove(p: Project): bool createProject(n: string): Project getProjects(): vector<Project> } ProjectService "1" *-- "1" Project </pre>
<p>При использовании отношения «часть-целое» требуется явно указать, сколько частей содержит целое – одну или несколько. Это выполняется указанием значений кратности ролей</p>	
<p>10.18 Использование при указании кратности роли неверного значения</p>	 <pre> classDiagram class СписокУровней { множествоУровней: Уровень[] количествоУровней: int СписокУровней() ~СписокУровней() добавитьУ(у: Уровень): void } class Уровень { описаниеУ: string баллУ: int Уровень(онУ: string, б: int) getОнУ(): string getБалл(): int } СписокУровней "1" o-- "-2..*" Уровень </pre>
<p>При указании кратности роли все числа должны быть целыми неотрицательными</p>	
<p>10.19 Использование при указании кратности роли неверного диапазона значений</p>	 <pre> classDiagram class СписокУровней { множествоУровней: Уровень[] количествоУровней: int СписокУровней() ~СписокУровней() добавитьУ(у: Уровень): void } class Уровень { описаниеУ: string баллУ: int Уровень(онУ: string, б: int) getОнУ(): string getБалл(): int } СписокУровней "1" o-- "5..2" Уровень </pre>
<p>При указании кратности роли в виде диапазона (промежутка) он должен начинаться с меньшего значения</p>	
<p>10.20 Отсутствие обобщения между классом, содержащим атрибут или операцию с областью видимости protected, и его потомками</p>	 <pre> classDiagram class ЗадачаИсполняемая { началоРаботы: time_t ЗадачаИсполняемая(нЗ: string, нР: time_t) getНР(): time_t {isQuery} редактироватьЗадачуИсполняемую1(нР: time_t): void } class ЗадачаНовая { названиеЗадачи: string ЗадачаНовая(нЗ: string) getНЗ(): string {isQuery} редактироватьНовуюЗадачуН(нЗ: string): void } ЗадачаНовая -- > ЗадачаИсполняемая </pre>
<p>Область видимости protected у члена класса указывает, что он виден только самому классу и его потомкам, поэтому следует установить обобщение между классом и его потомками</p>	
<p>10.21 Неверные имена в ограничении, связанном с понятием</p>	 <pre> classDiagram class СекретноеСообщение { тема: String списокПолучателей: String[] текстСообщения: String пароль: byte[] СекретноеСообщение() getТ(): String getСП(): String[] getП(): byte[] } </pre>
<p>Имена в тексте ограничения, связанного с понятием, должны совпадать с именами соответствующих атрибутов этого понятия</p>	

Продолжение таблицы 10.1

1	2	
10.22 Неверный тип связи между элементами «класс» и «перечисление»		
Между классом и перечислением следует установить связь типа «зависимость»		
Семантические ошибки		
10.23 Использование типа данных, не соответствующего контексту предметной области		
Тип данных char используется для представления одного символа. Наименование проекта не может состоять из одного символа, поэтому указанный тип данных для этого атрибута является ошибкой		
10.24 Неверная область видимости члена класса	 <p style="text-align: center;">а)</p>	 <p style="text-align: center;">б)</p>
	 <p style="text-align: center;">в)</p>	
<p>Указание у атрибутов/операций неверной области видимости приводит к нарушению принципов объектно-ориентированного проектирования и программирования:</p> <p>а) инкапсуляции – атрибуты класса рекомендуется помещать в закрытой части класса, а доступ к ним обеспечить с помощью операций из открытой части этого класса,</p> <p>б) инкапсуляции и наследования – между классами в модели отношение «часть-целое», а не «предок-потомок», поэтому атрибуты класса-части следует поместить в закрытой части,</p> <p>в) инкапсуляции и наследования – кроме указанных в модели классов доступ ко всем их атрибутам смогут получить объекты других классов, но классы-потомки не унаследуют атрибут фИО; поэтому атрибут класса Персона следует поместить в защищенную часть класса, а атрибуты классов Администратор и Пользователь – в закрытую</p>		

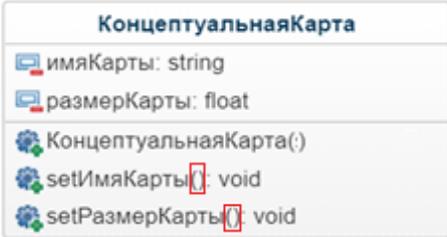
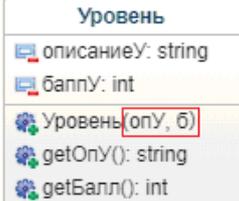
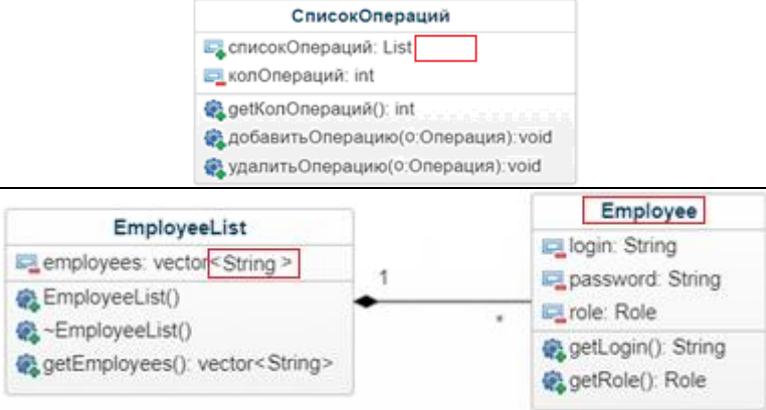
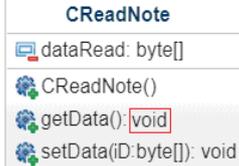
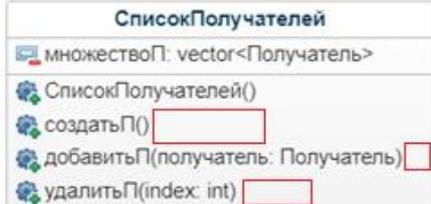
Продолжение таблицы 10.1

1	2	
<p>10.25 Лишний атрибут класса</p>		
<p>Для идентификации классов используются их уникальные имена. Такая диаграмма свидетельствует о том, что автор мыслит на уровне данных, а не на уровне модели</p>		
<p>10.26 Дублирование атрибутов у классов-потомков</p>		
<p>Ответы на вопросы «Что имеют общее?» и «Чем отличаются?» классы Пользователь, Отправитель и Администратор позволяют выявить их общие атрибуты: элАдрес и фИО. Поэтому эти два атрибута должны быть только у класса-предка Пользователь. Классы-потомки Отправитель и Администратор различаются между собой поведением. Также каждый из них имеет уникальные атрибуты статус и сервисНастройки соответственно</p>		
<p>10.27 Лишняя операция класса</p>		
<p>Использование лишних и семантически неверных атрибутов классов приводит к необходимости иметь в классе лишние операции для работы с такими атрибутами. Такая диаграмма свидетельствует о том, что автор мыслит на уровне данных, а не на уровне модели</p>		
<p>10.28 Неверный конструктор класса-потомка</p>		
<p>Недостаточно параметров у конструктора Администратор. Согласно принципу наследования конструктор класса-потомка предназначен для инициализации своих атрибутов и атрибутов класса-предка</p>		

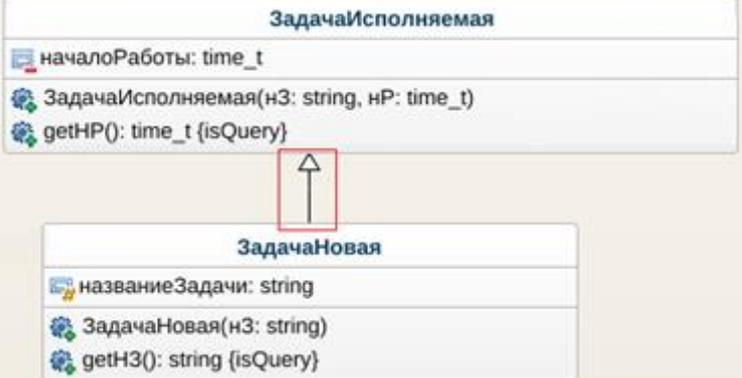
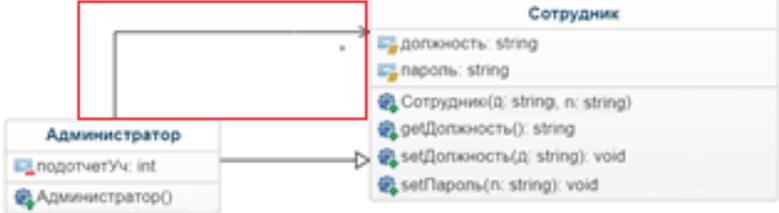
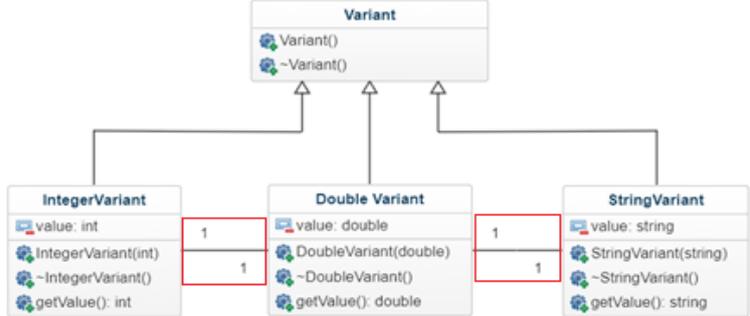
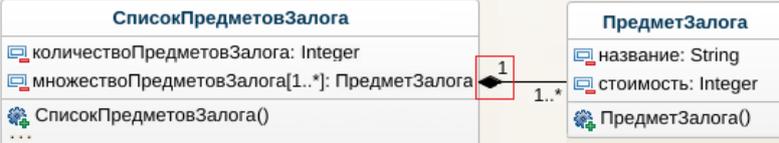
Продолжение таблицы 10.1

1	2
<p>10.29 Дублирование операций у классов-потомков</p>	
<p>Ответы на вопросы «Что имеют общее?» и «Чем отличаются?» классы Пользователь, Отправитель и Администратор позволяют выявить их общую операцию: создатьПисьмо. Поэтому эта операция должна быть у класса-предка Пользователь</p>	
<p>10.30 Отсутствие операций с типом данных класс-часть у класса-целого с атрибутом, отличным от стандартного контейнера языка ООП</p>	
<p>Если у класса-целого есть атрибут, отличный от стандартного контейнера языка ООП, то у такого класса обязательно должны быть операции (кроме конструктора/деструктора) для работы с элементами (типа класс-часть) этого атрибута. Отсутствие операций добавления / удаления элементов в атрибут множествоЗаявок является ошибкой</p>	
<p>10.31 Недопустимая операция класса</p>	
<p>Если у класса-целого есть атрибут, выполняющий роль контейнера объектов другого класса, то такому классу следует иметь операцию получения актуального значения количества таких объектов (get... для соответствующего атрибута количество...). Наличие операции установки значения такого атрибута (set...) является ошибкой, т.к. это значение количества объектов изменяют операции добавления / удаления элементов</p>	
<p>10.32 Лишний параметр у операции get...</p>	
<p>Операция get... предназначена для получения значения соответствующего атрибута, поэтому указание параметра у этой операции является ошибкой</p>	

Продолжение таблицы 10.1

1	2
10.33 Отсутствие параметров у операции set...	
<p>Операция set... предназначена для установки значения соответствующего атрибута, поэтому отсутствие параметра у этой операции является ошибкой</p>	
10.34 Отсутствие типа данных аргумента в описании операции	
<p>В описании операции для каждого ее аргумента следует указать имя и тип данных, разделяя их двоеточием. Отсутствие типа данных не позволяет проверить правильность передачи данных в такую операцию</p>	
10.35 Отсутствие или указание неверного имени типа данных класса-части у атрибута класса-целого	
<p>При композиции или агрегации в классе-целое следует у одного из атрибутов типом данных указать имя типа класса-части. Отсутствие у последовательного контейнера List типа данных его элементов не позволит ему хранить множество объектов класса-часть, что является ошибкой. Указание у последовательного контейнера vector типа String не позволит ему хранить множество объектов класса Employee, что является ошибкой</p>	
10.36 Неверный тип результата у операции get...	
<p>Операция get... предназначена для получения значения соответствующего атрибута, поэтому указание типа данных результата у get..., отличного от типа данных этого атрибута, является ошибкой</p>	
10.37 Отсутствие типа результата у операции (кроме конструкторов и деструкторов)	

Продолжение таблицы 10.1

1	2
	<p>У операции класса следует указать тип результата. Только операции «конструктор» и «деструктор» не должны иметь указания типа данных результата. Если операция возвращает только управление, то в языках программирования для обозначения этого предусмотрен тип void</p>
<p>10.38 Неверное направление у обобщения</p>	
<p>При таком направлении обобщения класс <i>ЗадачаНовая</i> кроме собственного атрибута <i>названиеЗадачи</i> и собственной операции <i>getНЗ</i>, получает атрибут <i>началоРаботы</i> и операцию <i>getНР</i>, что противоречит описанию ПроО</p>	
<p>10.39 Лишняя ассоциация между классом-предком и классом-потомком</p>	
<p>При наличии обобщения между двумя классами (предок-потомок) других структурных отношений между ними не должно быть</p>	
<p>10.40 Лишняя ассоциация между классами-потомками</p>	
<p>В иерархии обобщения классы-потомки могут иметь структурные отношения только с классом-предком</p>	
<p>10.41 Неверный вид агрегации</p>	
<p>Класс-часть может существовать без класса-целого, поэтому следует указать коллективную агрегацию</p>	

Продолжение таблицы 10.1

1	2
10.42 Неверное направление у агрегации / композиции	
При указании отношения «часть-целое» ромб должен быть размещен на стороне «целое»	
10.43 Неверная кратность агрегации (должна быть «один-ко-многим»)	
<p>Если атрибутом класса-целого в агрегации / композиции является класс-часть, то кратность роли класса-части должна быть больше единицы. Если атрибутом класса-целого в агрегации / композиции является класс-часть, то кратность роли класса-целого должна быть равна 1</p>	
10.44 Неверная кратность агрегации (должна быть «один-к-одному»)	
<p>Если атрибутом класса-целого в агрегации / композиции «частью» является класс-контейнер, то кратности ролей класса-целого и класса-части должны быть равны единице</p>	
10.45 «Цикл» в агрегации / композиции между классами	
<p>Класс-целое СписокОпераций не может быть одновременно «частью» самого себя (класс Операция)</p>	

11. Создание диаграмм компонентов

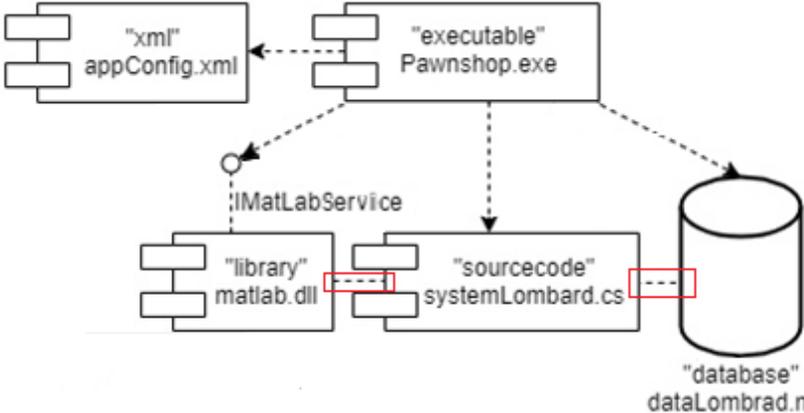
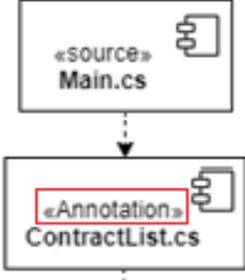
Диаграммы компонентов как модели этапа реализации используют для представления архитектуры ИС. Поэтому важна полная согласованность решений, зафиксированных:

- в концепции системы и принятых при выявлении требований к ИС,
- на этапе проектирования при выборе целевого языка ООП.

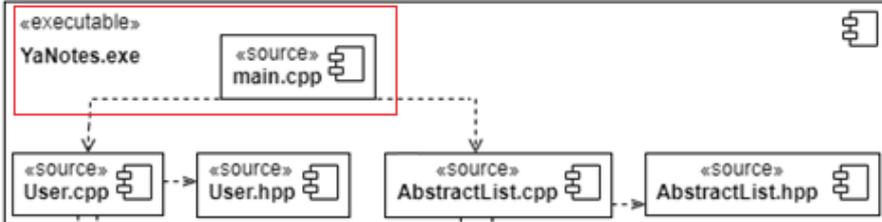
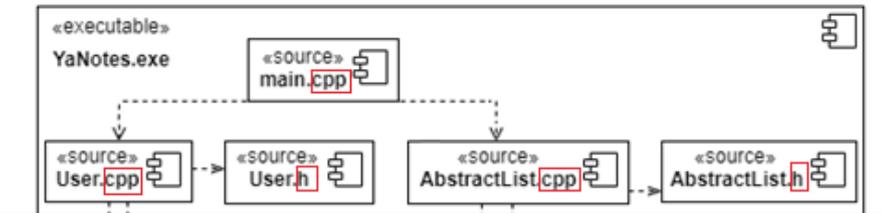
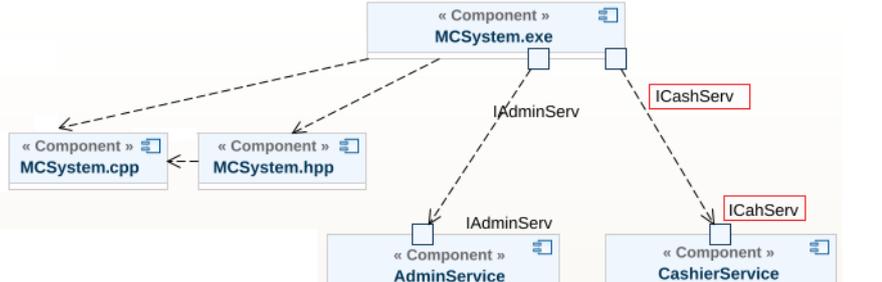
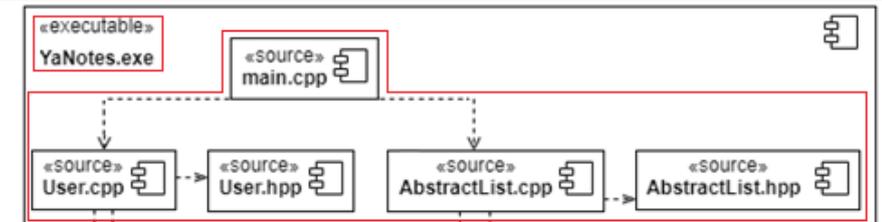
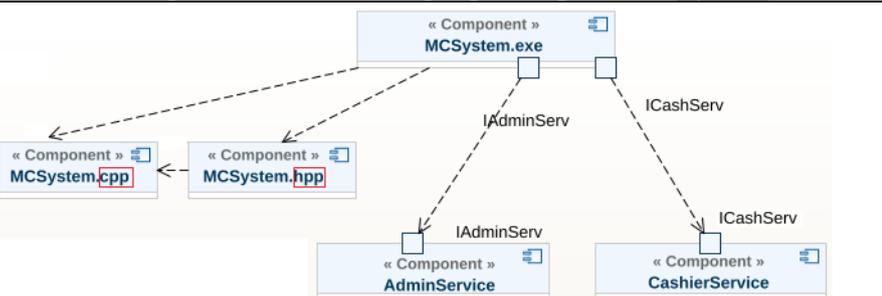
Для создания диаграмм компонентов следует использовать сервис genMyModel.

В таблице 11.1 представлены типичные ошибки, допускаемые студентами при создании CompD. Фрагменты диаграмм, которые представлены в этой таблице, могут содержать более одной ошибки в соответствующей строке; рассматриваемая ошибка выделена серым фоном текста, а на изображениях – красным контуром.

Таблица 11.1 – Ошибки в диаграммах компонентов

1	2
<p>11.1 Неверный символ зависимости</p>	<p style="text-align: center;">Лексические ошибки</p> 
	<p>Для изображения зависимости между компонентами следует использовать пунктирную стрелку</p>
<p>11.2 Использование стереотипов, которые не являются лексемами для диаграммы компонентов</p>	
<p>Стереотип «Annotation» не определен в языке UML</p>	

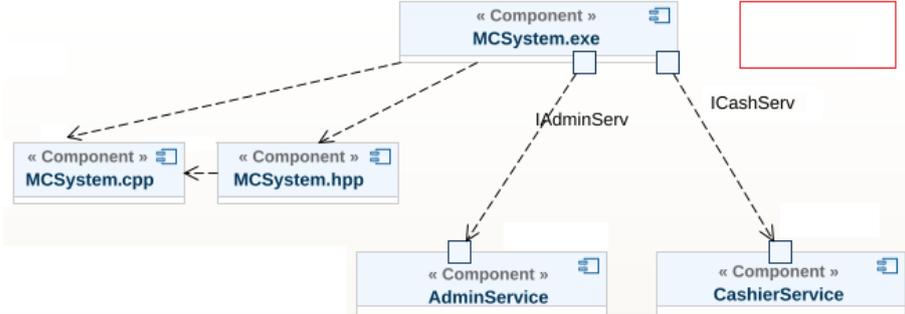
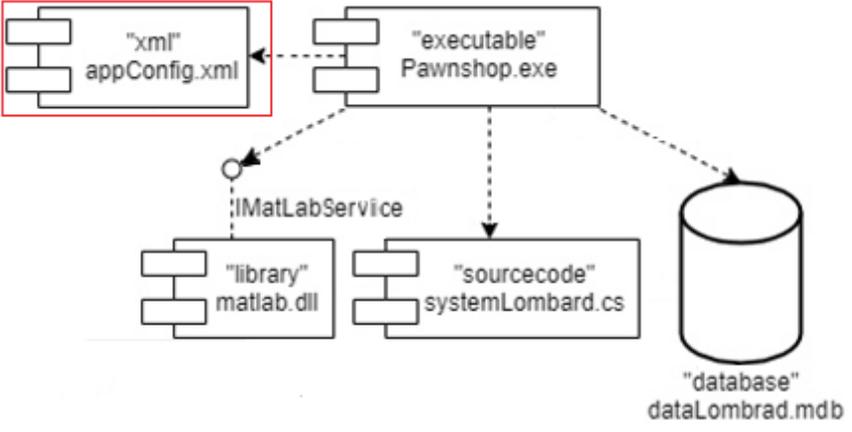
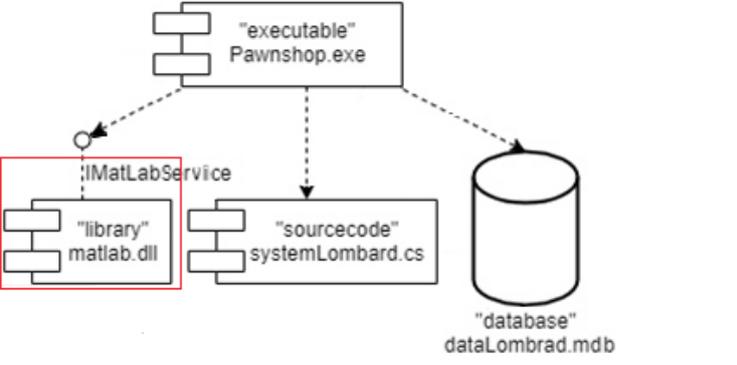
Продолжение таблицы 11.1

1	2
Синтаксические ошибки	
11.3 Отсутствие зависимости между компонентами	
Компонент исполняемого файла зависит от компонентов исходного кода	
11.4 Неверная информация о типе компонента исходного кода	<p>Текст отчета о создании диаграммы классов: «Для реализации системы использован язык программирования C++»</p> 
Смешение компонентов исходного кода двух языков программирования	
11.5 Отсутствие согласованности имен в интерфейсе	
Имена у предоставляемого и требуемого интерфейсов должны совпадать	
Семантические ошибки	
11.6 Размещение компонентов исходного кода внутри компонента исполняемого файла	
Компоненты исходного кода не могут быть составными частями компонента исполняемого файла	
11.7 Несоответствие типа компонента исходного кода языку ООП, использованному на этапе проектирования	<p>Текст отчета о создании диаграммы классов: «В системе применяется язык программирования Python»</p> 
Несоответствие типа компонента исходного кода языку ООП, использованному в CD	

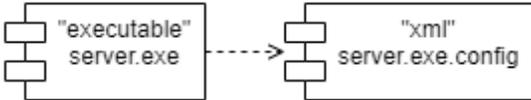
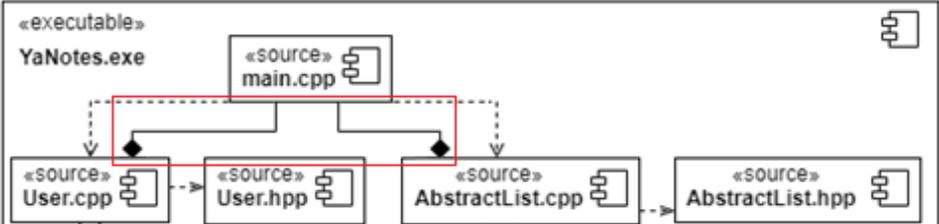
Продолжение таблицы 11.1

1	2
11.8 Отсутствие соответствия между диаграммой классов и диаграммой компонентов	
а)	
б)	
<p>Размещение договоров, существующих в ИС на различных стадиях их жизненного цикла, в различных компонентах (ошибка 11.8, а) противоречит обобщению, спроектированному в диаграмме классов (ошибка 11.8, б)). Также в компонентах не представлены предмет залога и их набор</p>	

Продолжение таблицы 11.1

1	2
<p>11.9 Отсутствие компонента базы данных</p>	<p>Текст отчета о концепции ИС: «4. Нефункциональными требованиями к MCSystem являются: – требования к производительности – ИС должна поддерживать не менее 50 одновременно работающих с БД пользователей»</p> 
<p>Для соответствия архитектуры системы и ее проектной организации следует включить компонент БД в модель этапа реализации</p>	
<p>11.10 Лишний компонент файла конфигурации</p>	<p>Текст отчета о концепции ИС: «4.4 Проектная организация Система должна быть интегрирована с существующей системой реестра договоров, функционирующей на основе реляционной СУБД»</p> 
<p>В проектной организации ИС не предусматриваются ее дополнительные настройки и их сохранение</p>	
<p>11.11 Лишний компонент библиотеки</p>	
<p>Использование пакета прикладных программ решения задач технических вычислений для выполнения расчетов по заключению, продлению и закрытию договоров ломбарда является избыточным</p>	

Продолжение таблицы 11.1

1	2
<p>11.12 Отсутствие компонентов клиентской части у ИС с веб-ориентированной архитектурой</p>	<p>Текст отчета о концепции ИС: «4.2 Удобство использования Программный интерфейс должен быть веб-ориентированным. Разрабатываемая система должна быть кроссбраузерной»</p>  <pre> graph LR subgraph Executable S1[executable server.exe] end subgraph XML S2["xml" server.exe.config] end S1 -.-> S2 </pre>
<p>Архитектура системы ограничена серверной частью</p>	
<p>11.13 Неверное использование агрегации</p>	 <pre> graph TD subgraph Executable YaNotes[«executable» YaNotes.exe] end subgraph SourceFiles main[«source» main.cpp] UserC[«source» User.cpp] UserH[«source» User.hpp] AbstractC[«source» AbstractList.cpp] AbstractH[«source» AbstractList.hpp] end YaNotes --> main YaNotes --> UserC YaNotes --> AbstractC UserC -.-> UserH AbstractC -.-> AbstractH </pre>
<p>Отношение «часть-целое» неверно использовано для компонентов исходного кода</p>	

12. Создание диаграмм развертывания студентов

В UML существует два вида диаграмм развертывания [5]:

- 1) описательная диаграмма развертывания, на которой изображены типы узлов системы и те компоненты, которые на них содержатся,
- 2) диаграмма развертывания в реальном мире, на которой показаны отдельные узлы и их связи в конкретной версии системы.

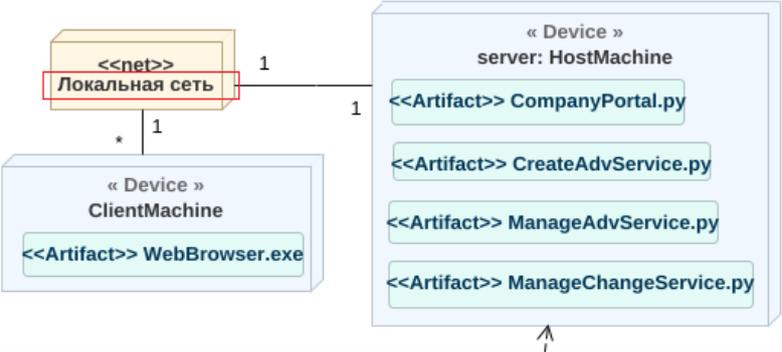
Для любого из этих видов важна согласованность DD:

- с решениями, зафиксированными в концепции системы и принятыми при выявлении требований к ИС,
- с диаграммой компонентов.

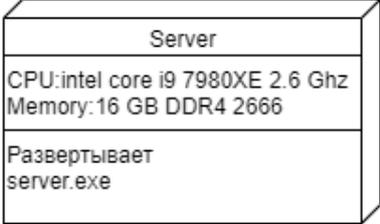
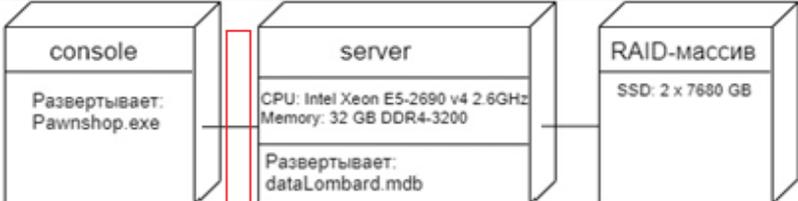
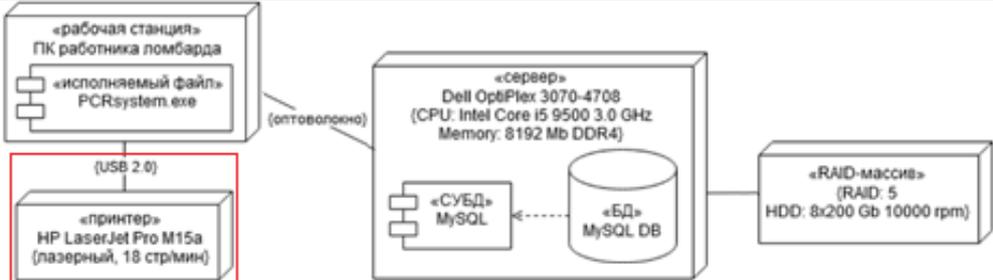
Для создания диаграмм компонентов следует использовать сервис genMyModel.

В таблице 12.1 представлены типичные ошибки, допускаемые студентами при создании DD. Фрагменты диаграмм, которые представлены в этой таблице, могут содержать более одной ошибки в соответствующей строке; рассматриваемая ошибка выделена серым фоном текста, а на изображениях – красным контуром.

Таблица 12.1 – Ошибки в диаграммах развертывания

1	2
<p>12.1 Указание локальной компьютерной сети вместо глобальной</p>	<p>Текст отчета с требованиями к ИС: «Название ПрО: Интернет-портал с информацией о предприятиях»</p>  <p>The diagram shows a deployment architecture. On the left, there is a node labeled «net» with the name «Локальная сеть» (Local network) inside. This node is connected to a «Device» named «ClientMachine» which contains an artifact «WebBrowser.exe». The «net» node is also connected to another «Device» named «server: HostMachine» which contains four artifacts: «CompanyPortal.py», «CreateAdvService.py», «ManageAdvService.py», and «ManageChangeService.py». The «net» node is highlighted with a red border, indicating an error.</p>
<p>Выбранный вид компьютерной сети противоречит описанию предметной области</p>	<p>↑</p>

Продолжение таблицы 12.1

1	2
<p>12.2 Отсутствие соответствия требований к ИС и типа приложения на диаграмме развертывания</p>	<p>Текст отчета с требованиями к ИС: "...ИС должна выдерживать нагрузку не менее 50 пользователей 1. Пользовательский интерфейс ИС должен быть веб-ориентированным..."</p>
	<p>a)</p> 
	<p>Текст отчета с требованиями к ИС: «4.3 Производительность Система должна поддерживать до 1000 одновременно работающих пользователей»</p>
<p>б)</p> 	
<p>Узлы диаграммы развертывания противоречат веб-ориентированной архитектуре: – в а) представлена только клиентская часть, – в б) представлена только серверная часть.</p>	
<p>12.3 Отсутствие компьютерной сети у системы с архитектурой «клиент-сервер»</p>	<p>Текст отчета с требованиями к ИС: «4.3 Производительность Система должна поддерживать до 100 человек, одновременно работающих в сети ломбарда»</p> 
<p>Архитектура ИС противоречит требованиям к этой системе</p>	
<p>12.4 Лишнее периферийное устройство</p>	
<p>Принтер не может быть узлом диаграммы развертывания, т.к. не является устройством, предоставляющим услуги (передачи или хранения данных)</p>	

13. Выписки из учебных планов

В таблице 13.1 представлены выписки из учебных планов с данными по дисциплине «Моделирование информационных систем».

Таблица 13.1 – Выписки из учебных планов

Наименование	Кол-во семестров	Лекции	Практические	Лабораторные работы	Самостоятельная работа	Кредиты	Форма отчетности, триместр
01.03.02 Прикладная математика и информатика	1	28	14	28	110	5	Экзамен, 10
10.05.01 Компьютерная безопасность	1	14	-	28	66	3	Зачет, 7

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 7.32–2017. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления: межгосударственный стандарт. М.: Стандартинформ, 2017. 32 с.
2. ГОСТ Р 7.0.5–2008 Национальный стандарт Российской Федерации. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая ссылка. Общие требования и правила составления. М.: Стандартинформ, 2008. 44 с.
3. About the Unified Modeling Language Specification Version 2.4. URL: <https://www.omg.org/spec/UML/2.4/> (дата обращения: 12.05.2022).
4. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. СПб: Питер, 2002. 496 с.
5. Рамбо Дж., Якобсон А., Буч Г. UML: Специальный справочник. СПб.: Питер, 2002. 656 с.
6. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. М.: ДМК Пресс, 2006. 496 с.
7. Nugroho, A., Flaton, B., Chaudron, M.R.V. Empirical Analysis of the Relation between Level of Detail in UML Models and Defect Density / MODELS 2008. 2008. Lecture Notes in Computer Science. Vol 5301. P. 600–614. URL: https://doi.org/10.1007/978-3-540-87875-9_42 (дата обращения: 12.05.2022).
8. Does the level of detail of UML diagrams affect the maintainability of source code?: a family of experiments / A.M. Fernández-Sáez [et al.] // Empirical Software Engineering. 2016. Vol. 21, No. 1. P. 212-259.
9. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции: В 2 т. М.: Мир, 1978. Т.1. 612 с.
10. Компиляторы. Принципы, технологии и инструментарий / А. Ахо [и др.]. М.: Диалектика, 2019. 768 с.
11. Основы проектирования и реализации информационных систем: учеб.-метод. пособие / Н. Н. Дацун [и др.]; ПГНИУ. Пермь, 2021. 132 с. URL: <http://www.psu.ru/files/docs/science/books/uchebnie-posobiya/Dacun-Otinov-Gasheva-Vlasov-Osnovy-roektirovaniya-i-realizacii-informacionnyh-sistem.pdf> (дата обращения: 12.05.2022).
12. GenMyModel. URL: <https://www.genmymodel.com/> (дата обращения: 12.05.2022).
13. Шаврин С.М., Лядова Л.Н., Чуприна С.И. Моделирование и проектирование информационных систем: учеб.-метод. пособие. Пермь: Изд-во Перм. гос. ун-та, 2007. 152 с.
14. Новиков Ф.А. Анализ и проектирование на UML: учебно-метод. пособие. СПб.: Санкт-Петерб. гос. ун-т ИТМО, 2007. 286 с. URL: <https://books.ifmo.ru/file/pdf/424.pdf> (дата обращения: 12.05.2022).
15. draw.io. URL: <https://app.diagrams.net/> (дата обращения: 12.05.2022)

Учебное издание

Дацун Наталья Николаевна

**Моделирование информационных систем.
Практикум**

Учебно-методическое пособие

Редактор *Л. Л. Савенкова*
Корректор *Л. Л. Соболева*
Компьютерная верстка: *Н. Н. Дацун*

Объем данных 1,25 Мб
Подписано к использованию 14.07.2022

Размещено в открытом доступе
на сайте www.psu.ru
в разделе НАУКА / Электронные публикации
и в электронной мультимедийной библиотеке ELiS

Издательский центр
Пермского государственного
национального исследовательского университета
614990, г. Пермь, ул. Букирева, 15