

**Санкт-Петербургский государственный университет
информационных технологий, механики и оптики**



**Описание самостоятельной работы студентов (СРС)
«Автоматизированные методы разработки
архитектуры программного обеспечения»**

Генельт А.Е.,
ассистент кафедры математического моделирования

**Санкт-Петербург
2007**

Введение

Целью самостоятельной работы по дисциплине «Автоматизированные методы разработки архитектуры ПО» является: приобретение студентами практических навыков выполнения всех частей проекта разработки программного продукта, включая анализ предметной области, разработки спецификации требований, моделирования ПО средствами с применением средств автоматизации, а также сбор и анализ атрибутов качества процесса и продукта разработки. При разработке архитектурной части проекта ПО необходимо использовать язык моделирования UML. Поскольку реальное проектирование выполняется коллективами проектировщиков, практические работы выполняются группами студентов.

Группы студентов, состоящие из не менее, чем 6-ти студентов разрабатывают и защищают следующие артефакты:

- 1) Протокол встречи с заказчиком (Текст 1-3 стр);
- 2) Одностраничное описание (Текст 1-3 стр);
- 3) Спецификация требований (ТЗ) (Текст, диаграммы использования, измеряемые показатели качества);
- 4) Детальный проект архитектуры (Текст, диаграммы всех типов, образы экранных форм, формулы, алгоритмы, детализированные измеряемые показатели качества, документация);
- 5) Прототип приложения (Экранные формы, код прототипа, пользовательская документация, исполняемый модуль на носителе);
- 6) Отчет по качеству проекта. (План и протоколы тестирования, обзоры кода (стиль), анализ значений измеренных показателей качества кода и документации);
- 7) Приемосдаточные испытания. (Презентация, протокол приемосдаточных испытаний).

Группы студентов выбирают задания проектов ПО из следующих примеров предметной области:

- Предметная область, выбранная группой студентов;
- Программное обеспечение банкомата;
- Программное обеспечение мобильного телефона;
- Программное обеспечение музыкального центра;
- Информационная система библиотеки;

- Информационная система поликлиники;
- Информационная система деканата;
- Система мгновенного обмена сообщениями;
- Информационная система склада;
- Система учета рабочего времени;
- Информационная система жилищного агентства;
- Информационная система технической экспертизы;
- Система продажи билетов на футбол.

Выполнение работ производится с применением доступных средств автоматизации архитектурного проектирования.

Наиболее удобным и распространенным средством автоматизации проектирования архитектуры ПО является CASE-средство IBM Rational Rose. CASE-система IBM Rational Software - Rational Rose позволяет автоматизировать этапы анализа и проектирования разрабатываемого программного обеспечения, а также предоставляет возможность генерации кода ПО на различных языках программирования для формирования макетов систем и позволяет автоматизировать выпуск проектной документации.

Rational Rose позволяет разрабатывать проектную документацию в виде диаграмм и спецификаций, а также производить генерацию программного кода на различных языках программирования (C++, Smalltalk, PowerBuilder, Ada, SQLWindows и ObjectPro). В составе инструментальных программных средств CASE-системы Rational Rose, также содержатся средства реинжиниринга ПО. Такая возможность предназначена для повторного использования программных компонент в новых проектах.

Методической основой применения CASE-системы Rational Rose является автоматизация процесса построения диаграмм классов, состояний, сценариев, модулей и процессов, а также формализации спецификаций логической и физической структуры модели, и описания статических и динамических аспектов разрабатываемого ПО.

Уникальность CASE-системы Rational Rose заключается в обеспечении архитектора ПО (проектировщика ПО) достаточными средствами проектирования, в том числе: репозиторий, графический интерфейс, средства просмотра проекта, средства контроля проекта, средства сбора статистики и генератор документов, генератор и анализатор программного кода и средства реинжиниринга.

Репозиторий CASE-системы Rational Rose обеспечивают "навигацию" по проекту (включая перемещение по иерархиям классов и подсистем, переключение от одного вида диаграмм к другому, средства контроля и сбора статистики, генератор отчетов и др.) позволяют моделировать проект ПО и сопровождать результат разработки в течение всего жизненного цикла программной системы.

Создаваемый встроенным генератором CASE-системы Rational Rose скелет кода программы на языке программирования С++ предназначен для его доработки традиционным методом прямого программирования на языке С++. При генерации программного кода в CASE-системе Rational Rose используется информация из логической и физической моделей проекта ПО. В результате “прогона” генератора формируются заголовки и описания классов и объектов.

Анализатор исходного кода С++ позволяет создавать модули проектов и осуществляет контроль правильности исходных текстов и диагностику ошибок. Получаемая модель проекта пригодна для её использования в качестве повторно применяемого кода.

CASE-система Rational Rose позволяет формировать такие проектные документы, как:

- диаграммы классов;
- диаграммы состояний;
- диаграммы сценариев;
- диаграммы модулей;
- диаграммы процессов;
- спецификации классов, объектов, атрибутов и операций
- заготовки текстов программ,

а также модель разрабатываемой программной системы в текстовом формате (.mdl-файл).

Одним из лучших пособий по практическому применению IBM Rational Rose, изданных на русском языке, является работа Уэнди Боггса (Wendy Boggs) и Майкла Боггса (Michael Boggs) – “UML и Rational Rose”, выпущенная в переводе к. т. н. Александра Михайловича Вендрова [1].

Приведённый здесь материал частично основывается на данной книге, но для полноценного выполнения работ все же необходимо наличие оригинального Пособия.

Создание диаграммы Вариантов Использования (Use Case)

Создание диаграммы Вариантов Использования и действующих лиц

Создайте диаграмму вариантов использования для системы обработки заказов. Готовая диаграмма Вариантов Использования должна выглядеть как на рисунке 1.

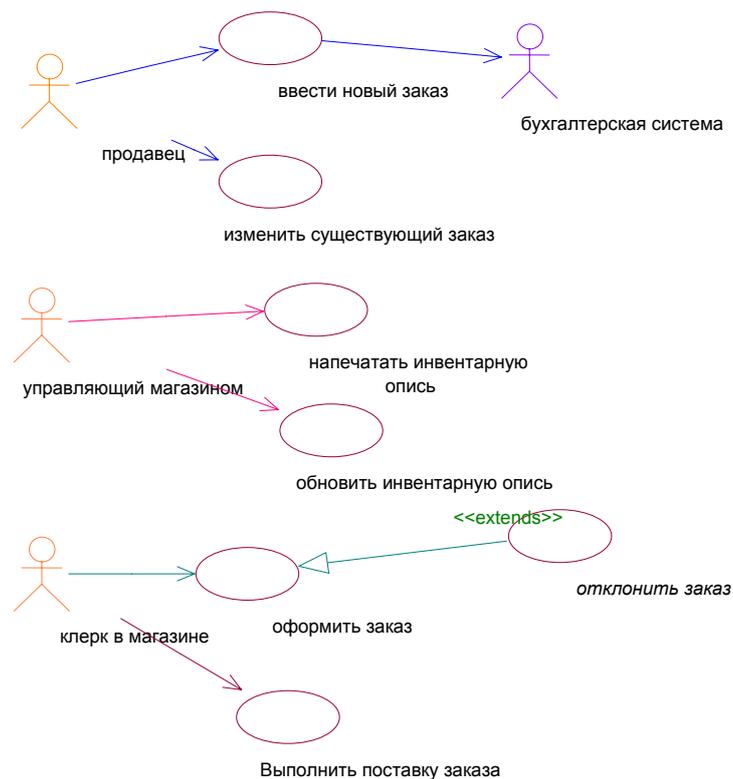


Рисунок 1. Диаграмма Вариантов Использования для системы обработки заказов

Этапы выполнения упражнения

- 1) Откройте главную диаграмму Вариантов Использования (Main) в браузере;
- 2) Используя инструмент Use Case в панели инструментов, поместите на диаграмму новый вариант использования с идентификатором "Ввести новый заказ";
- 3) Поместите на диаграмму остальные варианты использования:
 - "Изменить существующий заказ"
 - "Напечатать инвентарную опись"
 - "Обновить инвентарную опись"

- “Оформить заказ”
 - “Отклонить заказ”
 - “Выполнить поставку заказа”
- 4) Используя инструмент Actor (Действующее лицо) панели инструментов поместите на диаграмму новое действующее лицо "Продавец"
- 5) Поместите на диаграмму остальных действующих лиц:
- “Управляющий магазином”
 - “Клерк магазина”
 - “Бухгалтерская система”
- 6) Создайте абстрактный вариант использования. Для этого на Use Case диаграмме "Отклонить заказ" установите флажок Abstract в пункте меню Open Specification (Открыть спецификацию).
- 7) Выполните добавление ассоциаций, нарисуйте ассоциацию между действующим лицом “Продавец” и вариантом использования "Ввести заказ" используя кнопки Unidirectional Association (Однонаправленная ассоциация) из панели инструментов.
- 8) Поместите на диаграмму остальные ассоциации, как показано на рис. 1.
- 9) Выполните добавление связи расширения с помощью кнопки Generalization (Обобщение) панели инструментов. Для этого нарисуйте связь между вариантом использования "Отклонить заказ" и вариантом использования "Оформить заказ". Направление стрелки от первого варианта использования ко второму. Такая связь расширения означает, что вариант использования "Отклонить заказ" дополняет функциональные возможности варианта использования "Оформить заказ". Используйте правую кнопку мыши на добавленной связи между вариантами использования "Отклонить заказ" и "Оформить заказ". Выберите пункт Open Specification (Открыть спецификацию). В списке стереотипов введите “extends” (расширение), затем нажмите <ОК>. В результате этого действия на линии данной связи появится надпись “extends”.
- 10) Добавьте описания к Use Case диаграмме. Выделите в браузере Use Case "Ввести новый заказ" и введите следующее описание: "Этот вариант использования дает клиенту возможность ввести новый заказ в систему" в окне документации. С помощью окна документации добавьте описания ко всем остальным вариантам использования.
- 11) Добавьте описания к действующему лицу. Для этого выделите в браузере действующее лицо “Продавец”. Введите следующее описание: "Продавец — это служащий, старающийся продать товар" в окне документации. Так же введите описания к остальным действующим лицам.

Создание диаграмм взаимодействия

В этом упражнении будут разработаны диаграммы Последовательности и Кооперативные диаграммы, описывающие введение нового заказа в нашу систему обработки заказов.

Создание диаграммы Последовательности ввода нового заказа после завершения первого этапа работы

Создайте диаграмму Последовательности и Кооперативную диаграмму, отражающую ввод нового заказа в систему обработки заказов. Готовая диаграмма Последовательности должна выглядеть так:

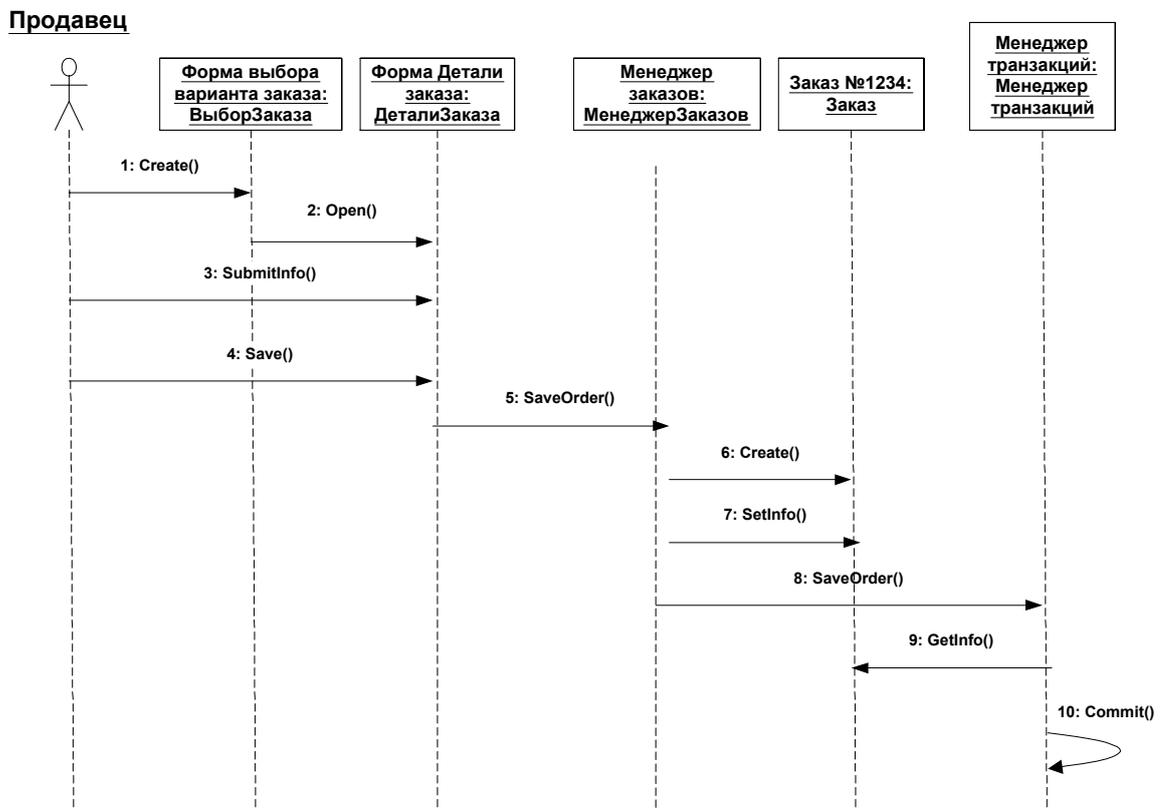


Рисунок 2. Диаграмма Последовательности

Уэнди Боггс (Wendy Boggs) и Майкл Боггс (Michael Boggs) так комментируют эту диаграмму: “...Это только одна из диаграмм, необходимых для моделирования варианта использования "Ввести новый заказ". Она соответствует успешному варианту хода событий. Для описания того, что случится, если возникнет ошибка, или если пользователь выберет другие действия из предложенных, придется разработать другие диаграммы. Каждый альтернативный поток варианта использования может быть промоделирован с помощью своих собственных диаграмм Взаимодействия”.

Настройка программной среды

- 1) Выберите пункт Tools > Options (Инструменты > Параметры) в меню модели.
- 2) Установите флажки Sequence Numbering, Collaboration Numbering и Focus of Control на вкладке диаграмм и нажмите <ОК>.

Создание диаграммы Последовательности

- 3) В меню логического представления броузера выберите пункт New > Sequence Diagram.
- 4) Назовите новую диаграмму "Ввод заказа" и откройте её.

Добавление на диаграмму действующего лица и объектов

- 5) Перетащите на диаграмму действующее лицо Продавец (Salesperson) с панели броузера.
- 6) Нажмите кнопку Object (Объект) на панели инструментов.
- 7) Щелкните мышью в верхней части диаграммы, для помещения в неё нового объекта и назовите объект "Выбор варианта заказа".
- 8) Аналогично поместите на диаграмму все остальные объекты ("Форма Детали заказа" и номер заказа - "Заказ №1234").

Добавление сообщений на диаграмму

- 9) Нажмите кнопку Object Message (Сообщение объекта) на панели инструментов.
- 10) Выделив сообщение, введите имя сообщения - "Создать новый заказ".
- 11) Поместите на диаграмму также сообщения: "Открыть форму" между "Выбор Варианта Заказа" и "Форма деталей Заказа".
- 12) Повторите эти же шаги для сообщений:
 - – "Ввести номер заказа, заказчика и число заказываемых предметов" — между "Продавец" и "Форма Деталей Заказа";
 - – "Сохранить заказ" — между "Продавец" и "Форма Деталей Заказа";
 - – "Создать пустой заказ" — между "Форма Деталей Заказа" и "Заказ N1234";
 - – "Ввести номер заказа, заказчика и число заказываемых предметов" — между "Форма Деталей Заказа" и "Заказ N1234".
 - – "Сохранить заказ" — между "Форма Деталей Заказа" и "Заказ N1234".

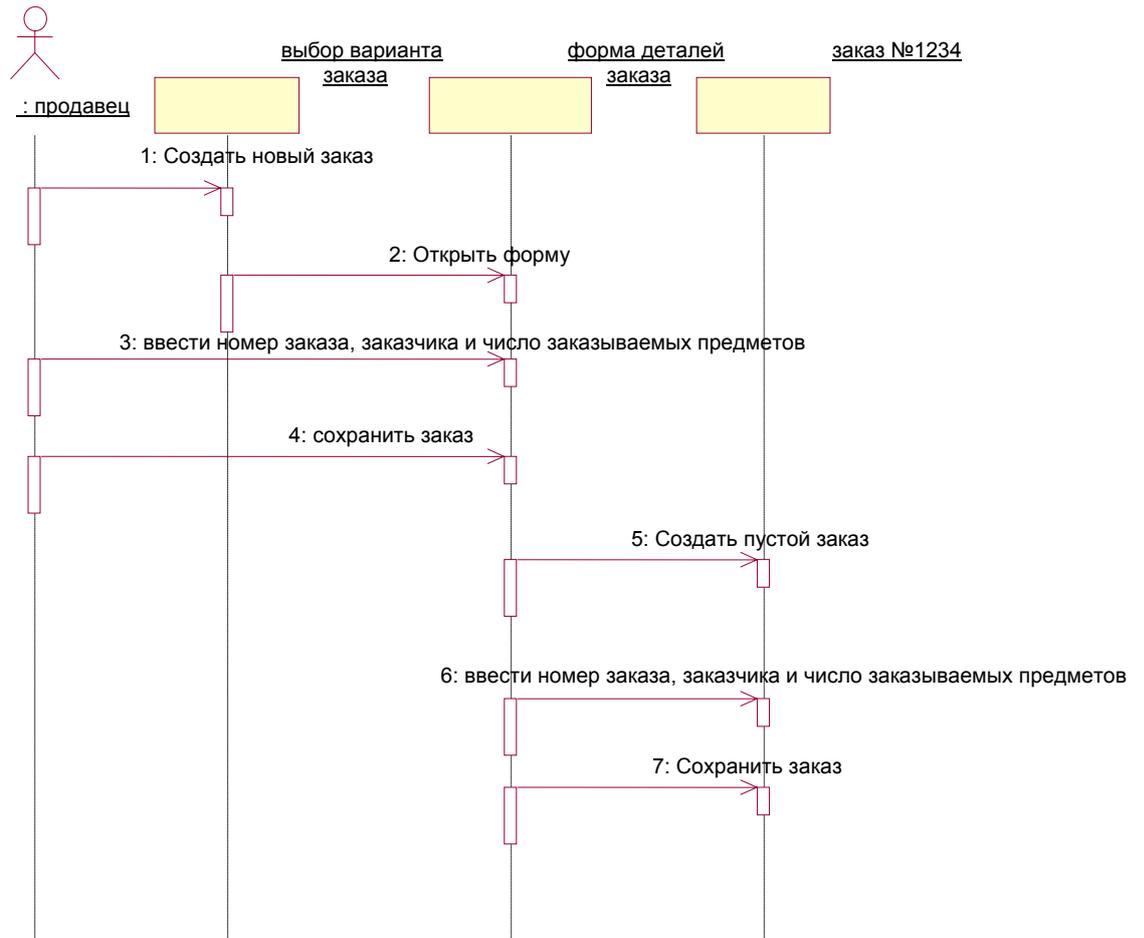


Рисунок 3. Диаграмма Последовательности ввода нового заказа после завершения текущего этапа работы

Теперь надо позаботиться об управляющих объектах и взаимодействии с базой данных. Как видно из диаграммы, объект Детали заказа имеет множество ответственностей, с которыми лучше всего мог бы справиться управляющий объект. Кроме того, новый заказ должен сохранять себя в базе данных сам. Вероятно, эту обязанность лучше было бы переложить на другой объект.

Добавление на диаграмму дополнительных объектов

- 13) Для помещения нового объекта нажмите кнопку Object панели инструментов и щелкните мышью между объектами “Форма Деталей Заказа” и “Заказ №1234”. Введите имя объекта “Управляющий заказами”.
- 14) Нажмите кнопку Object панели инструментов. Расположите новый объект правее “Заказ №1234”. Введите имя объекта “Управляющий транзакциями”.

Назначение ответственностей объектам

- 15) Выделите текст “Создать пустой заказ” и удалите этот текст, при помощи <CTRL>+<D>.
- 16) Аналогичным образом удалите сообщения “Вести номер заказа, заказчика и число заказываемых предметов” и “Сохранить заказ”.
- 17) Поместите на диаграмму новое сообщение - “Сохранить заказ”, расположив его под сообщением 4 между “Форма деталей заказа” и “Управляющий заказами”, используя кнопку Object Message панели инструментов.
- 18) Добавьте аналогичным образом следующие сообщения:
 - – “Создать новый заказ” — между “Управляющий заказами” и “Заказ №1234”;
 - – “Ввести номер заказа, заказчика и число заказываемых предметов” - между “Управляющий заказами” и “Заказ №1234”;
 - – “Сохранить заказ” - между “Управляющий заказами” и “Управляющий транзакциями”;
 - – “Информация о заказе” — между “Управляющий транзакциями” и “Заказ №1234”.
- 19) Затем нажмите кнопку Message to Self (Сообщение себе) на панели инструментов, щелкните на линии жизни объекта “Управляющий транзакциями” несколько ниже сообщения “Информация о заказе – с одноименной операцией”, добавив туда рефлексивное сообщение ”Сохранить информацию о заказе в базе данных”.

Соотнесение объектов с классами

- 20) Укажите мышью на объект “Выбор варианта заказа”, и щелкните правой кнопкой мыши. Выберите пункт Open Specification (Открыть спецификацию) в открывшемся меню.
- 21) Выберите пункт <New> (Создать) в списке классов, в результате чего оявится окно спецификации классов.
- 22) Введите “Выбор заказа” в поле Name <ОК>. Система вернется в окно спецификации объекта.
- 23) Выберите класс “Выбор Заказа” в списке классов. Для возврата к диаграмме нажмите <ОК>. Созданный объект называется “Выбор варианта заказа: Выбор Заказа”.
- 24) Аналогичным образом соотнесите остальные классы с объектами:
 - Класс “Детали заказа” с объектом “Форма деталей заказа”;
 - Класс “Упр_заказами” с объектом “Управляющий заказами”;
 - Класс “Заказ” с объектом “Заказ N 1234”;

- Класс “Упр_ транзакциями” с объектом “Управляющий транзакциями”.

Соотнесение сообщений с операциями

- 25) Щелкните правой кнопкой мыши на объекте “Создать новый заказ”, и выберите пункт <new operation> (создать операцию) в открывшемся меню. Должно появиться окно спецификации операции.
- 26) Введите имя операции ”Создать поле Name” <ОК>.
- 27) Аналогично елкните правой кнопкой мыши на объекте “Создать новый заказ” и в открывшемся меню выберите новую операцию “Создать()”.
- 28) Повторите эти действия, чтобы соотнести с операциями все остальные сообщения:
 - “Открыть форму” соотнесите с операцией “Открыть()”;
 - “Ввести номер заказа, заказчика и число заказываемых предметов” — с операцией “Ввести номер заказа, заказчика и число заказываемых предметов()”;
 - “Сохранить заказ” — с операцией “Сохранить заказ()”;
 - “Создать пустой заказ” – с операцией “Создать пустой заказ()”;
 - “Ввести номер заказа, заказчика и число заказываемых предметов” - с одноименной операцией;
 - “Сохранить заказ” – с операцией “Сохранить заказ()”;
 - “Информация о заказе: – с одноименной операцией;
 - “Сохранить информацию о заказе” - с одноименной операцией.

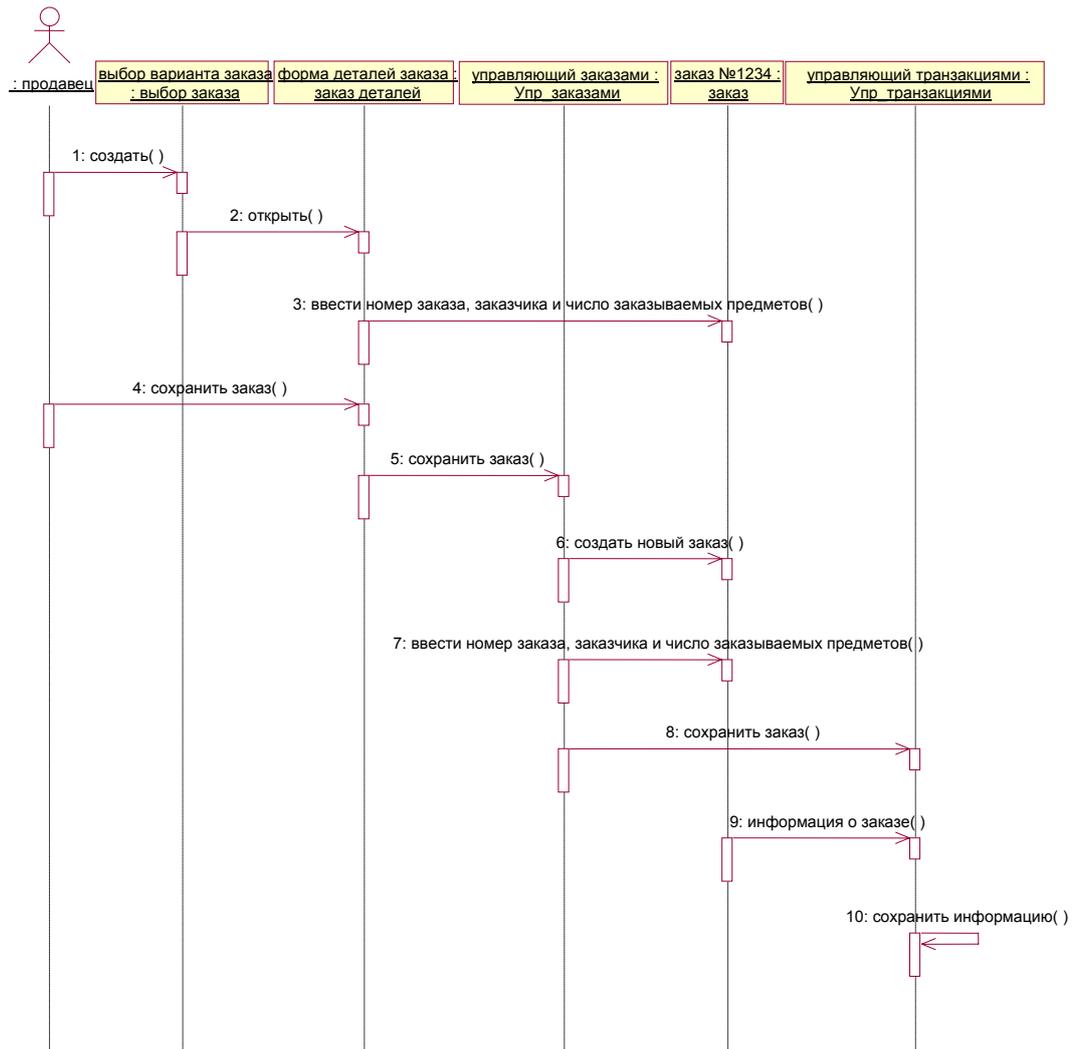


Рисунок 4. Вид диаграммы последовательности

Создание Кооперативной диаграммы

Целью дальнейших действий должна быть кооперативная диаграмма (диаграмма кооперации).

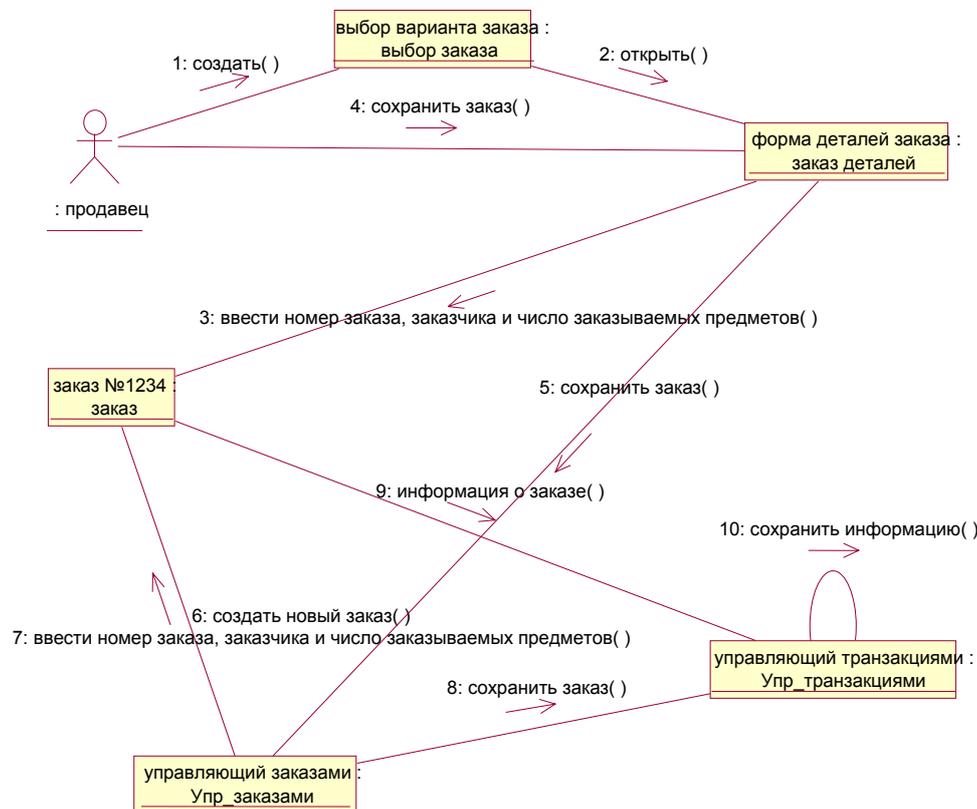


Рисунок 5. Вид диаграмма кооперации

- 29) Щелкните правой кнопкой мыши на Логическом представлении в браузере. Выберите пункт New > Collaboration Diagram (Создать > Кооперативная диаграмма) в открывшемся меню и назовите эту диаграмму “Ввод заказа”.
- 30) Откройте эту диаграмму для добавления действующего лица и объектов на диаграмму.
- 31) Перетащите действующее лицо “Продавец” из браузера на диаграмму.
- 32) Кликните кнопку Object (Объект) панели инструментов, затем кликните мышью внутри диаграммы для помещения в неё нового объекта “Выбор варианта заказа”.
- 33) Аналогично поместите на диаграмму объекты: “Форма деталей заказа” и “Заказ №1234”.

Добавление сообщений на диаграмму

- 34) Нажмите кнопку Object Link (Связь объекта) на панели инструментов.
- 35) Проведите мышью от действующего лица “Продавец” по направлению к объекту “Выбор варианта заказа”.
- 36) Выполните аналогичные операции для назначения связей следующих объектов:

“Действующее лицо Продавец” и объект “Форма деталей Заказа”;
“Объект Форма деталей Заказа” и объект “Выбор Варианта Заказа”;
“Объект Форма деталей Заказа” и объект “Заказ N1234”.

- 37) Нажмите кнопку Link Message (Сообщение связи) на панели инструментов и щелкните мышью по связи “Продавец” и “Форма деталей Заказа”.
- 38) Выделив сообщение, введите имя “Создать новый заказ”;
- 39) Выполните аналогичные операции для помещения на диаграмме сообщений:
 - “Открыть форму” между “Выбор Варианта Заказа” и “Форма Деталей Заказа”;
 - “Ввести номер заказа, заказчика и число заказываемых предметов” между “Продавец” и “Форма Деталей Заказа”;
 - “Сохранить заказ” между “Продавец” и “Форма деталей Заказа”;
 - “Создать пустой заказ” между “Форма деталей Заказа” и “Заказ №1234”;
 - “Ввести номер заказа, заказчика и число заказываемых предметов” между “Форма деталей Заказа” и “Заказ №1234”;
 - “Сохранить заказ” между “Форма деталей Заказа” и “Заказ №1234”

Добавление дополнительных объектов на диаграмму

- 40) Нажмите кнопку Object на панели инструментов и щелкните мышью на диаграмме, для помещения нового объекта с именем “Управляющий заказами”.
- 41) Аналогично поместите на диаграмме еще один объект - “Управляющий транзакциями”.

Добавление дополнительных объектов на диаграмму

- 42) Выделите сообщение “Создать пустой заказ”, выделяя, при этом слова, а не стрелку.
- 43) Удалите это сообщение (<CTRL>+<D>).
- 44) Выполните аналогичные операции для удаления сообщений: “Ввести номер заказа, заказчика и число заказываемых предметов” и “Сохранить заказ”.
- 45) Выделите связь между объектами “Форма деталей Заказа” и “Заказ №1234”, удалите и эту связь.
- 46) На панели инструментов нажмите кнопку Object Link (Связь объекта).
- 47) Нарисуйте связь между Форма деталей Заказа и Управляющий Заказа-ми.
- 48) На панели инструментов нажмите кнопку Object Link (Связь объекта).
- 49) Нарисуйте связь между “Управляющий Заказами” и “Заказ №1234”.
- 50) На панели инструментов нажмите кнопку Object Link (Связь объекта).
- 51) Нарисуйте связь между “Заказ №1234” и “Управляющий Транзакцией”.
- 52) Нажмите кнопку Object Link (Связь объекта) на панели инструментов и нарисуйте связь между “Управляющий Заказами” и “Управляющий Транзакцией”.
- 53) Нажмите кнопку Link Message (Сообщение связи) на панели инструментов, щелкните мышью на связи между объектами “Форма деталей Заказа” и “Управляющий Заказами” для ввода нового сообщения “Сохранить заказ”.
- 54) Аналогично введите сообщения относительно имеющихся сообщений, назвав их:
 - “Создать пустой заказ” — между “Управляющий Заказами” и “Заказ №1234”;
 - “Ввести номер заказа, заказчика и число заказываемых предметов” — между “Управляющий Заказами” и “Заказ №1234”;
 - “Сохранить заказ” — между “Управляющий Заказами” и “Управляющий Транзакцией”
 - “Информация о заказе” — между “Управляющий Транзакцией” и “Заказ №1234”
- 55) Нажмите кнопку Link to Self (Связь с собой) на панели инструментов и, щелкнув на объект “Управляющий Транзакцией”, добавьте к нему рефлексивное сообщение. Нажмите кнопку Link Message (Сообщение связи). и, щелкнув на по рефлексивной связи

“Управляющий Транзакциями”, введите ”Сохранить информацию о заказе в базе данных”.

Соотнесение объектов с классами - 2

- 56) Создайте требующиеся классы “Выбор Заказа”, “Заказ деталей”, “ Упр_заказами”, “Заказ” и “Упр_транзакциями” так, как это делалось для диаграммы последовательностей.
- 57) Найдите в браузере класс “Выбор Заказа” и перетащите его на объект “Выбор варианта заказа” на диаграмме.
- 58) Аналогично поступите с объектами и соответствующими им классами:
- класс “Заказ деталей” соотнесите с объектом “Форма деталей заказа”;
 - класс “Упр_заказами” — с объектом “Управляющий Заказами”;
 - класс “Заказ” — с объектом “Заказ №1234”;
 - класс “Упр_транзакциями” — с объектом “Управляющий транзакциями”.

Соотнесение объектов с классами – 3

- 59) Правой кнопкой мыши щлкните на объекте “Форма деталей Заказа”. Выберите пункт Open Specification (Открыть спецификацию). В раскрывающемся списке классов выберите пункт <New> (Создать) и введите “Выбор заказа” < ОК > в поле имени спецификации классов. В списке классов выберите класс “Выбор заказа”< ОК >. Полученный объект называется “Выбор варианта заказа: Выбор Заказа”.
- 60) Аналогично выполните соотнесение остальных объектов с классами:
- класс “Детали заказа” — с объектом “Форма деталей заказа”;
 - класс “Упр_заказами” — с объектом “Управляющий заказами”;
 - класс “Заказ” — с объектом “Заказ N 1234”;
 - класс “Упр_ транзакциями” — с объектом “Управляющий транзакциями”.

Соотнесение сообщений с операциями - 2

- 61) Щелкните мышью по сообщению “Создать новый заказ”, выберите пункт Open Specification (Открыть спецификацию) и в раскрывающемся списке имен укажите имя операции — “Создать()”<ОК>.
- 62) Аналогично поступите с операциями остальных сообщений:
- “Открыть форму” соотнести с операцией “Открыть()”;
 - “Ввести номер заказа, заказчика и число заказываемых предметов” соотнести с операцией “Ввести номер заказа, заказчика и число заказываемых предметов()”;
 - “Сохранить заказ” соотнести с операцией “Сохранить заказ()”;
 - “Сохранить заказ” соотнести с операцией “Сохранить заказ()”;
 - “Создать пустой заказ” соотнести с операцией “Создать пустой заказ()”;
 - “Ввести номер заказа, заказчика и число заказываемых предметов” соотнести с одноименной операцией;
 - “Сохранить заказ” соотнести с операцией “Сохранить заказ()”;
 - “Информация о заказе” соотнести с одноименной операцией;
 - “Сохранить информацию о заказе” соотнести с одноименной операцией.

Соотнесение сообщений с операциями - 3

- 63) Щелкните мышью по сообщению “Создать новый заказ”, выберите пункт <New operation> (создать операцию). и в поле имени укажите имя операции — “Создать()”<ОК>.
- 64) Опять кликните мышью по сообщению “Создать новый заказ” и в открывшемся меню выберите пункт Open Specification (Открыть спецификацию) и в раскрывающемся списке Name (Имя) укажите имя новой операции <ОК>.
- 65) Аналогично поступите с операциями остальных сообщений:
- “Открыть форму” соотнести с операцией “Открыть()”;
 - “Ввести номер заказа, заказчика и число заказываемых предметов” соотнести с операцией “Ввести номер заказа, заказчика и число заказываемых предметов()”;
 - “Сохранить заказ” соотнести с операцией “Сохранить заказ()”;
 - “Сохранить заказ” соотнести с операцией “Сохранить заказ()”;
 - “Создать пустой заказ” соотнести с операцией “Создать пустой заказ()”;
 - “Ввести номер заказа, заказчика и число заказываемых предметов” соотнести с одноименной операцией;

“Сохранить заказ” соотнести с операцией “Сохранить заказ()”;
“Информация о заказе” соотнести с одноименной операцией;
“Сохранить информацию о заказе соотнести с одноименной операцией.

Диаграмма Состояний для класса "Заказ"

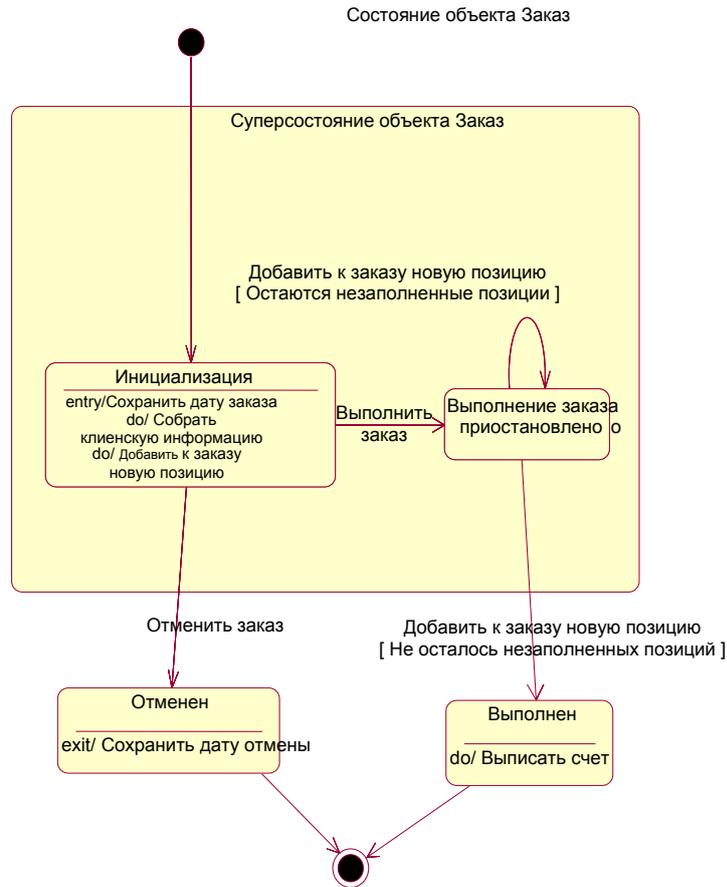


Рисунок 6. Диаграмма состояний для класса "Заказ"

- 66) Найдите в браузере класс "Заказ" и кликните на него правой кнопкой мыши. Укажите пункт <New > Statechart Diagram (Создать диаграмму состояний) в открывшемся меню.

Добавление начального и конечного состояний

- 67) Нажмите кнопку Start State (Начальное состояние) в панели инстру-ментов и поместите это состояние на диаграмму.
- 68) Нажмите кнопку End State (Конечное состояние) в панели инстру-ментов и поместите это состояние на диаграмму.

Добавление суперсостояния

69) Нажмите кнопку State (Состояние) в панели инструментов и поместите это состояние на диаграмму.

Добавление оставшихся состояний

70) Нажмите кнопку State (Состояние) в панели инструментов и поместите это состояние на диаграмму. Назовите состояние “Отменен”.

71) Нажмите кнопку State (Состояние) в панели инструментов и поместите это состояние на диаграмму. Назовите состояние ”Выполнен”.

72) Нажмите кнопку State (Состояние) в панели инструментов и поместите это состояние на диаграмму. Поместите состояние на диаграмму внутрь суперсостояния. Назовите состояние “Инициализация”.

73) Нажмите кнопку State (Состояние) в панели инструментов и поместите это состояние на диаграмму. Поместите состояние на диаграмму внутрь суперсостояния. Назовите состояние “Выполнение заказа приостановлено”.

Описание состояний

74) Двойным кликом мыши щелкните по состоянию “Инициализация”, перейдите на вкладку Detail (Подробно) и щелкните правой кнопкой мыши в окне Actions(Действия). Выберите пункт Insert(Вставить) в открывшемся меню.

75) Дважды щелкните мышью на новом действии и назовите его “Сохранить дату заказа”.

76) Убедитесь, что в окне When (Когда) указан пункт On Entry (На входе).

77) Аналогично добавьте следующие действия:

- “Собрать клиентскую информацию”. В окне When указать DO (Вы-полнять между входом и выходом);
- “Добавить к заказу новые позиции”. В окне When указать DO;

78) Нажмите два раза на <ОК>, чтобы закрыть спецификацию.

79) Дважды щелкните мышью на состоянии “Отменен”.

80) Аналогично добавьте “Сохранить дату отмены”, укажите - On Exit (На выходе);

81) Нажмите два раза на <ОК> для закрытия спецификации.

82) Дважды щелкните мышью на состоянии “Выполнен”.

- 83) Аналогично добавьте действие: “Выписать счет”, укажите On Exit.
- 84) Нажмите два раза на <ОК> для закрытия спецификации.

Добавление переходов

- 85) Используя кнопку Transition (Переход) панели инструментов, щелкните мышью на начальном состоянии. Проведите линию перехода к состоянию “Инициализация”.
- 86) Таким же образом создайте следующие переходы:
- от состояния “Инициализация” к состоянию “Выполнение заказа приостановлено”;
 - от состояния “Выполнение заказа приостановлено” к состоянию “Выполнен”;
 - от суперсостояния к состоянию “Отменен”;
 - от состояния “Отменен” к конечному состоянию;
 - от состояния “Выполнен” к конечному состоянию.
- 87) Используя кнопку Transition to Self (Переход к себе) панели инструментов, щелкните мышью по состоянию “Выполнение заказа приостановлено”

Описание переходов

- 88) Щелкните двойным кликом мыши по переходу от состояния “Инициализация” к состоянию “Выполнение заказа приостановлено”, откройте, таким образом, окно спецификации перехода.
- 89) Введите фразу “Выполнить заказ” поля Event (Событие) и <ОК > для закрытия окна спецификации.
- 90) Аналогичными действиями добавьте событие “Отменить заказ” к переходу между суперсостоянием и состоянием “Отменен”.
- 91) Щелкните двойным кликом мыши по переходу от состояния “Выполнение заказа приостановлено” к состоянию “Выполнен”, откройте окно его спецификации.
- 92) Введите фразу “Добавить к заказу новую позицию” поля Event (Событие) и < ОК > для закрытия окна спецификации.
- 93) Перейдите на вкладку Detail (Подробно).
- 94) Введите текст “Не осталось незаполненных позиций” в поле Guard Condition (Сторожевое Условие).
- 95) Щелкнув на <ОК>, закройте окно спецификации.

- 96) Щелкните двойным кликом мыши по рефлексивному переходу (Transition to Self) состояния “Выполнение заказа приостановлено”.
- 97) Введите текст “Добавить к заказу новую позицию” в поле Event (Событие).
- 98) Перейдите на вкладку Detail (Подробно).
- 99) Введите текст “Остаются незаполненные позиции” в поле Guard Condition (Сторожевое Условие).
- 100) Щелкнув на <ОК>, закройте окно спецификации.

Описание варианта использования.

При оформлении заказа проверяют наличие соответствующих товаров на складе. После этого выписываются товары для реализации заказа. Во время выполнения этих процедур одновременно проверяется прохождение платежей. Поставка осуществляется в случае наличия платежа и наличия товара на складе. Если платеж прошел, но товары на складе отсутствуют, то заказ ставится в положение ожидания. Если платеж не прошел, то заказ аннулируется.

- 101) Найдите в браузере вариант использования “Выполнить поставку заказа”, и меню правой кнопкой мыши. Укажите пункт New > Activity Diagram (Создать диаграмму активности (деятельности)). Назвите диаграмму “Выполнить поставку” и откройте ее двойным кликом мыши.
- 102) Щелкните мышью на элемент Swimlane панели инструментов и затем на поле диаграммы. На диаграмме появится разделительная линия (“водная дорожка”).
- 103) Установите курсор на заголовок NewSwimlane и нажмите правую клавишу мыши. Кликните Select in browser панели инструментов. В браузере, при этом, выделится объект. Кликните правой мышью клавишей на Open Specification и откройте спецификацию.
- 104) Измените поле Name на “Клерк”. Выберите в поле Class “Клерк в магазине”.
- 105) Аналогично создайте “Система”, Class – “Бухгалтерская система”.
- 106) Найдите в браузере сплошной черный кружок (начальное состояние) и перенесите его на дорожку “Клерк”.
- 107) Выберите объект Activity из панели инструментов и поместите его на диаграмму в “дорожку” “Клерк”. Измените имя объекта на ”Получить заказ”.
- 108) Аналогично создайте на “дорожке” “Клерк” новые Activity и присвойте им имена: “Проверить позицию заказа”, “Закрепить

- позицию за заказом”, “Поставить заказ в ожидание” и “Укомплектовать заказ”.
- 109) Поместите на “дорожку” новые объекты End State (конечное состояние). Измените значение поля Name одного из них на “Выполнить поставку”
- 110) Поместите на дорожку “Система” новый объект Activity и присвойте значение поля Name “Проверить платеж”. Поместите на эту же дорожку новый объект End State и измените в его спецификации значение поля Name на “Отменить заказ”.
- 111) Для установления связи объектов, поместите на “дорожку” “Клерк” два объекта Horizontal Synchronization (горизонтальная синхронизация) и назначьте значениям полей Name одного из объектов “1”, а другого - “2”.
- 112) Поместите на “дорожку” Клерк объект Decision (выбор). Присвойте полю Name значение “Позиция имеется?”.
- 113) Поместите на “дорожку” “Система” объект Decision. Присвойте полю Name значение “Деньги поступили?”.
- 114) Для соединения связей объектов, щелкните мышкой по объекту-стрелке State Transition (состояние перехода). Затем на начальное состояние объекта на диаграмме. Удерживая кнопку мыши, перенесите курсор на активность ”Получить заказ” и лишь затем освободите курсор. Два объекта будут соединены стрелкой.
- 115) Аналогично соедините:
- “Активность” “Получить заказ” с объектом Horizontal Synchronization “1”;
 - 1 и “Проверить платеж”;
 - 1 и “Проверить позицию заказа”;
 - “Деньги поступили?” и “Отменить заказ”;
 - “Позиция имеется” и “Закрепить позицию за заказом”;
 - “Позиция имеется?” и “Поставить заказ в ожидание”;
 - “Проверить заказ” и “Деньги подступили?”;
 - “Проверить позицию заказа” и “Позиция имеется”;
 - 2 и “Закрепить позицию за заказом”;
 - 2 и “Укомплектовать заказ”;
 - 2 и “Деньги получены?”;
 - “Поставить заказ в ожидание” и объект “Конечное состояние (без имени);
 - “Укомплектовать заказ” и “Выполнить поставку”.
- 116) Для присвоения наименования стрелок в поле Event (условие перехода), установим курсор на стрелке, соединяющей “Деньги получены?” и “Отменить заказ”. Двойным мышинным кликом откроем окно спецификации и вводим значение “Нет”, поля Event.
- 117) Аналогичным методом, для стрелки, соединяющей “Деньги получены?” и 2, вводим значение “Да”.

- 118) Для стрелки соединяющей “Позиция имеется?” и “Закрепить позицию за заказом” вводим “Да”.
- 119) Для стрелки между “Позиция имеется?” и “Поставить заказ в ожидание” вводим “Нет”.
- 120) Добавьте элементарные действия (Actions) к активности “Проверить позицию заказа”. Установите курсор на “Проверить позицию заказа” и двойным щелчком по клавише мыши откройте окно спецификации. Откройте закладку Actions. Установите курсор на свободное поле и нажмите на правую мышиную клавишу. В выпадающем меню кликните Insert. В появившейся заставке в поле When выберите Entry (на входе в активность), в поле Name введите текст “Просмотреть спецификацию к заказу” <Ok>.
- 121) Кликнув с помощью правой клавиши мыши, введите новое действие. Полю When присвойте Do (промежуток между входом и выходом), а полю Name присвойте “Найти новую позицию”.
- 122) При создании третьей активности полю When присвойте Exit (выход), а полю Name “Передать результаты поиска”.
- 123) За счет перемещения объектов можно привести диаграмму к следующему виду:

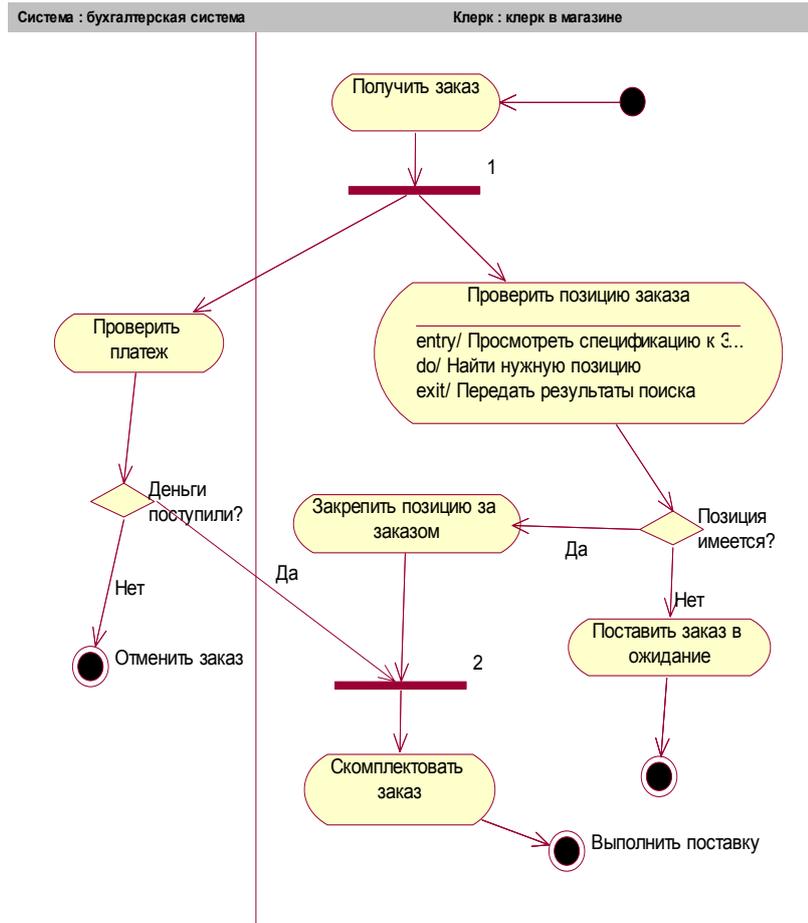


Рисунок 7. Диаграмма деятельности (активности) для варианта использования "Выполнить поставку заказа"

Изучение пройденного материала вполне достаточно для начального освоения IBM Rational Rose. Практическое освоение CASE-средства Rational Rose выполняйте с Пособием Уэнди Боггса и Майкла Боггса "UML и Rational Rose", выпущенным в переводе к. т. н. Александра Михайловича Вендрова. Наиболее интересными практическими частями Пособия являются упражнения с кодогенерацией проектов архитектуры ПО.

Рекомендуемая литература

М. Фаулер, К. Скотт. UML в кратком изложении. Применение стандартного языка объектного моделирования: Пер. с англ. – М.: Мир, 1999



Образец Технического задания для оформления самостоятельных работ студентов

**Техническое задание
по проекту
“sBS”**

Версия 1.0

Составлена:

Дата создания: 18.05.07

Содержание

Регистрация изменений

1. Введение
 - 1.1 Название программы
 - 1.2 Область применения
2. Основания для разработки
 - 2.1 Документы
 - 2.2 Организация утвердившая документы
 - 2.3 Сокращенное название проекта
3. Назначение разработки
 - 3.1 Проект предназначен для
 - 3.2 Система должна реализовывать
4. Основные требования к системе
 - 4.1 Общие функциональные требования
 - 4.2 Сущности с которыми работает система
 - 4.3 Основные функциональные роли пользователей системы
 - 4.4 Описание вариантов использования
 - 4.5 Нефункциональные требования
5. Особенности реализации
6. Требования к GUI
7. Стадии и этапы разработки
8. Глоссарий

«Автоматизированные методы разработки архитектуры ПО.
Описание самостоятельной работы студентов (СРС)»

Регистрация изменений

Изменение	Дата	Автор	Версия
Немного изменена структура требований в пункте 4.1, изменена регистрация изменений, исправлены орфографические ошибки, добавлены подписи к рисункам и схемам.	25.05.07	Шевченко Алексей	1.02

1. Введение.

1.1. Название программы - Биллинговая система для тарификации звонков клиентов оператора телефонии.

1.2. Область применения – программа используется для обеспечения, учета и тарификации звонков внутри системы телефонной связи.

2. Основания для разработки.

2.1. Разработка ведется на основании следующих документов: протокол первого совещания с участием заказчика, одностороннее описание проекта.

2.2. Организация, утвердившая документы – СПбГУ ИТМО. Дата утверждения – 17 мая 2007 года.

2.3. Сокращенное название проекта sBS (smart Billing System)

3. Назначение разработки

3.1. Проект предназначен для учебных целей, отработки навыков работы в команде, работы с блок схемами UML, а так же повышение общих навыков программирования и связей с СУБД.

3.2. Система должна реализовывать биллинговую систему локального оператора связи. Обеспечивается регистрация звонков, списание средств со счета клиента в соответствии с тарифом, а также ведение статистики.

4. Основные требования к системе

4.1. Общие функциональные требования

4.1.1. Сущности, с которыми работает система

Система работает со следующими сущностями:

- Клиент
- Счет
- Тариф
- Звонок
- Платеж

На Рис. 1 представлена диаграмма классов для сущностей системы.

«Автоматизированные методы разработки архитектуры ПО.
Описание самостоятельной работы студентов (СРС)»

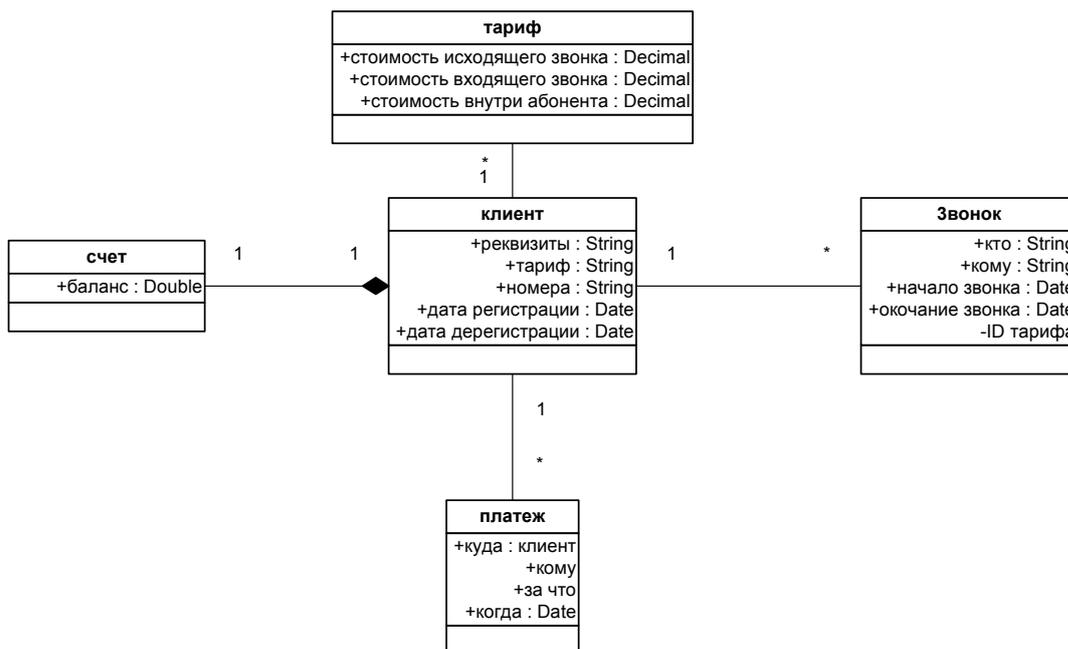


Рис. 1

- 4.1.2. Формат номера: максимальное число знаков – 8; первые два знака возможно латинские буквы.
- 4.1.3. *Тарификация*³ по разным тарифам.
- 4.1.4. Система обеспечивает ведение счетов *клиентов*.
- 4.1.5. Отчёты о звонках, платежах, состоянии счетов.
- 4.1.6. Информация о *клиентах*, звонках и *тарифах*² и *тарификации*³ хранится **вечно**.
- 4.1.7. Хранимые данные:
- Все стандартные реквизиты *клиентов*.
 - Информация обо **всех** платежах.
 - Информация о звонках, в том числе **не завершившихся**.
 - Все телефонные номера и информация о них.
- 4.1.8. *Тариф* содержит:
- **Три числа**: стоимость секунды разговора внутри *клиента*; стоимость секунды входящего и исходящего разговора между *клиентами*.
 - Все параметры *тарифа* указываются в рублях с точностью до копейки.
 - *Тарификация* ведётся в целых секундах, начиная с первой.
 - В каждый момент времени одному *клиенту* соответствует **только один тариф**.
 - *Клиент* может менять *тариф* за определённую плату.
 - Абонентской платы **нет**.
 - *Тарификация* происходит по окончании звонка.

- Оператор может закрыть существующий *тариф* или создать новый *тариф*.
- Менять параметры существующего *тарифа* **запрещено**.

4.1.9. Случай нулевого *баланса*⁴:

- Текущий звонок **не обрывается** даже в случае, когда *баланс клиента* меньше или равен нулю.
- Если *баланс клиента* меньше или равен нулю, то принимать звонки и звонить **запрещается**.

4.1.10. Счёт:

- Счёт составляется на одного *клиента*.
- Списание средств со счёта происходит по завершению звонка.
- Зачисление средств на счет производится **вручную** оператором.
- Зачисление средств на счет осуществляется по безналичному расчёту в рублях с точностью до копейки.

4.1.11. Отчёты:

- Оператору могут предоставляться отчеты как о *клиенте* в целом, так и об отдельных номерах.
- Отчёт формируется и высылается *клиенту* раз в месяц.
- Существует возможность получения *клиентом* отчёта по требованию.
- Возможна отправка отчёта по e-mail'у или факсу.
- Состав отчёта о *клиенте*: реквизиты *клиента*; суммарная плата за все его телефонные номера.
- Состав отчёта о номере: сам номер; плата за этот номер.
- Отчёты формируются автоматически не более 10 – 15 минут.

4.1.12. Возможности *клиента*:

- Блокировка / разблокировка телефонов внутри *клиента*.
- *Клиент* может менять *тариф* за определённую плату.
- *Клиент* по требованию может получить отчёт о себе.
- *Клиент* может просматривать совершённые и начатые звонки, ранжированные по длительности.

4.1.13. Возможности оператора:

- Блокировка / разблокировка как *клиентов* в целом, так и отдельных номеров.
- Оператор может закрыть существующий *тариф* или создать новый *тариф*.

- Зачислять средства на счёт **вручную** в зависимости от перечисленных *клиентом* средств через интерфейс системы.
- **Любое** изменение в базе данных протоколируется.
- Администратор/оператор может просмотреть список изменений базы данных.
- Оператору могут предоставляться отчеты как о *клиенте* в целом, так и об отдельных номерах.
- Оператор может производить резервное копирование **любых данных** базы.

4.1.14. Архив данных:

- Архивная информация это **вся** информация о *клиенте*, о его номерах и звонках, а так же о подписанных им *тарифах*.
- Архивная информация хранится **отдельно** от текущей информации.
- Архивирование информации (занос текущей информации в архив) происходит первый раз через три месяца после начала работы системы, далее **каждые три месяца** и хранится **вечно**.

4.1.15. Система по собственной инициативе **не прерывает** ни один из начатых звонков.

4.2. Основные функциональные роли пользователей системы.

4.2.1. Система предоставляет функциональность для следующих ролей пользователей: администратор, оператор, пользователь.

4.2.2. Администратору предоставляются следующие варианты использования системы:

- Регистрация тарифа
- Дерегистрация тарифа
- Управление учетными записями
- Формирование отчета

4.2.3. Оператору предоставляются следующие варианты использования системы:

- Регистрация нового клиента
- Дерегистрация клиента
- Занесение средств на счет
- Изменение режима обслуживания
- Формирование отчета

4.2.4. Пользователю предоставляются следующие варианты использования системы:

- Звонок

«Автоматизированные методы разработки архитектуры ПО.
Описание самостоятельной работы студентов (СРС)»

- Ответ на звонок
- Окончание разговора

4.3. Описание вариантов использования

4.3.1. Вариант использования “Регистрация клиента”. Действия системы при реализации данного варианта использования изображены на диаграмме деятельности Рис. 2

«Автоматизированные методы разработки архитектуры ПО.
Описание самостоятельной работы студентов (СРС)»

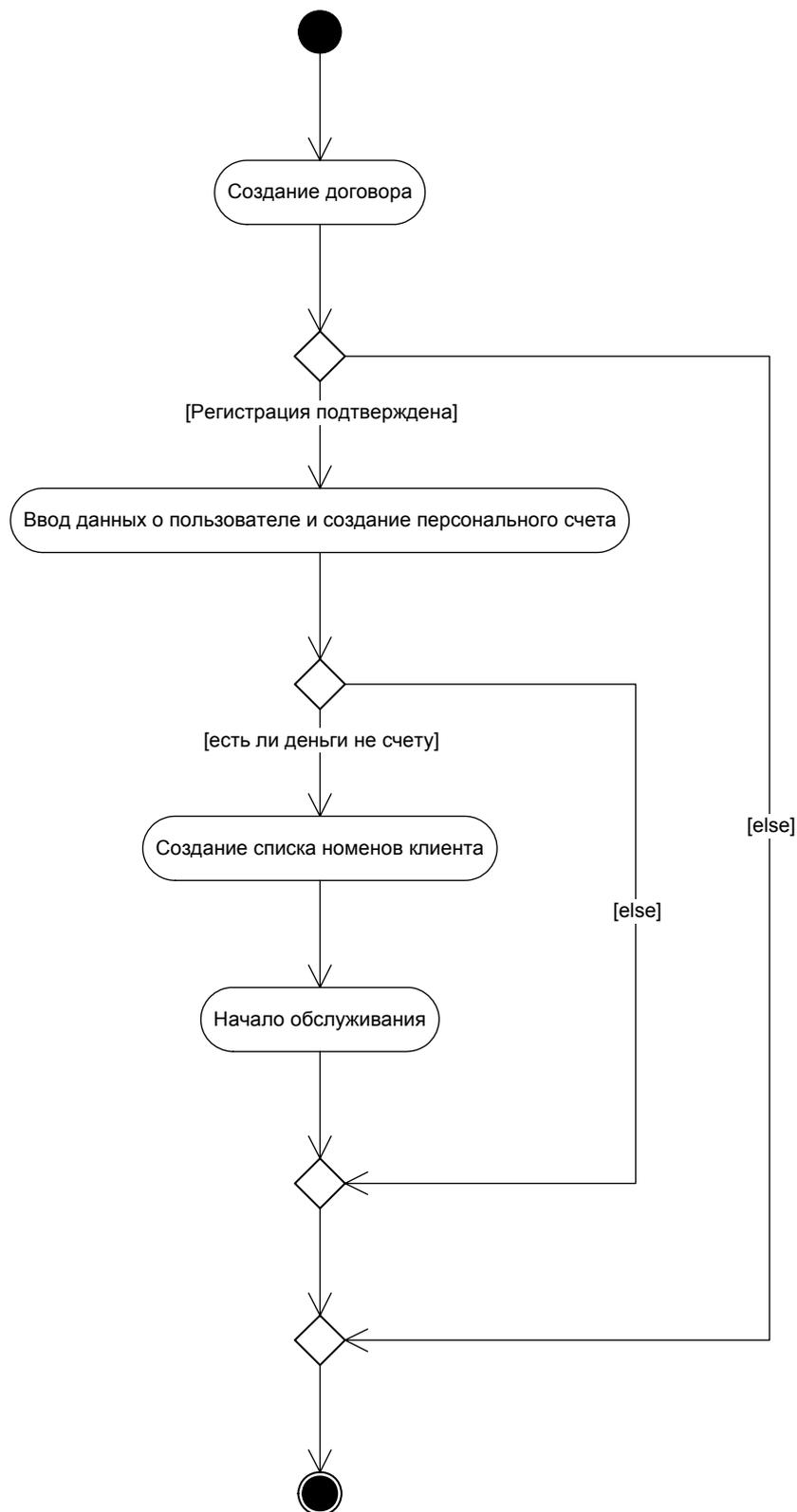


Рис. 2

Для выполнения этого варианта использования должны быть выполнены следующие условия:

- При регистрации клиента должны быть заданы все реквизиты клиента - юридического лица. Система не

«Автоматизированные методы разработки архитектуры ПО.
Описание самостоятельной работы студентов (СРС)»

позволяет зарегистрировать клиента при отсутствии хотя бы одного реквизита.

- При регистрации клиента, клиент должен иметь сумму денег для оплаты подключения
- Количество клиентов не должно превышать запланированный масштаб системы

При не выполнении этих условий, система должна генерировать соответствующие им исключения

4.3.2. Вариант использования “Дерегистрация клиента”. Действия системы при реализации данного варианта использования изображены на диаграмме деятельности Рис. 3

«Автоматизированные методы разработки архитектуры ПО.
Описание самостоятельной работы студентов (СРС)»

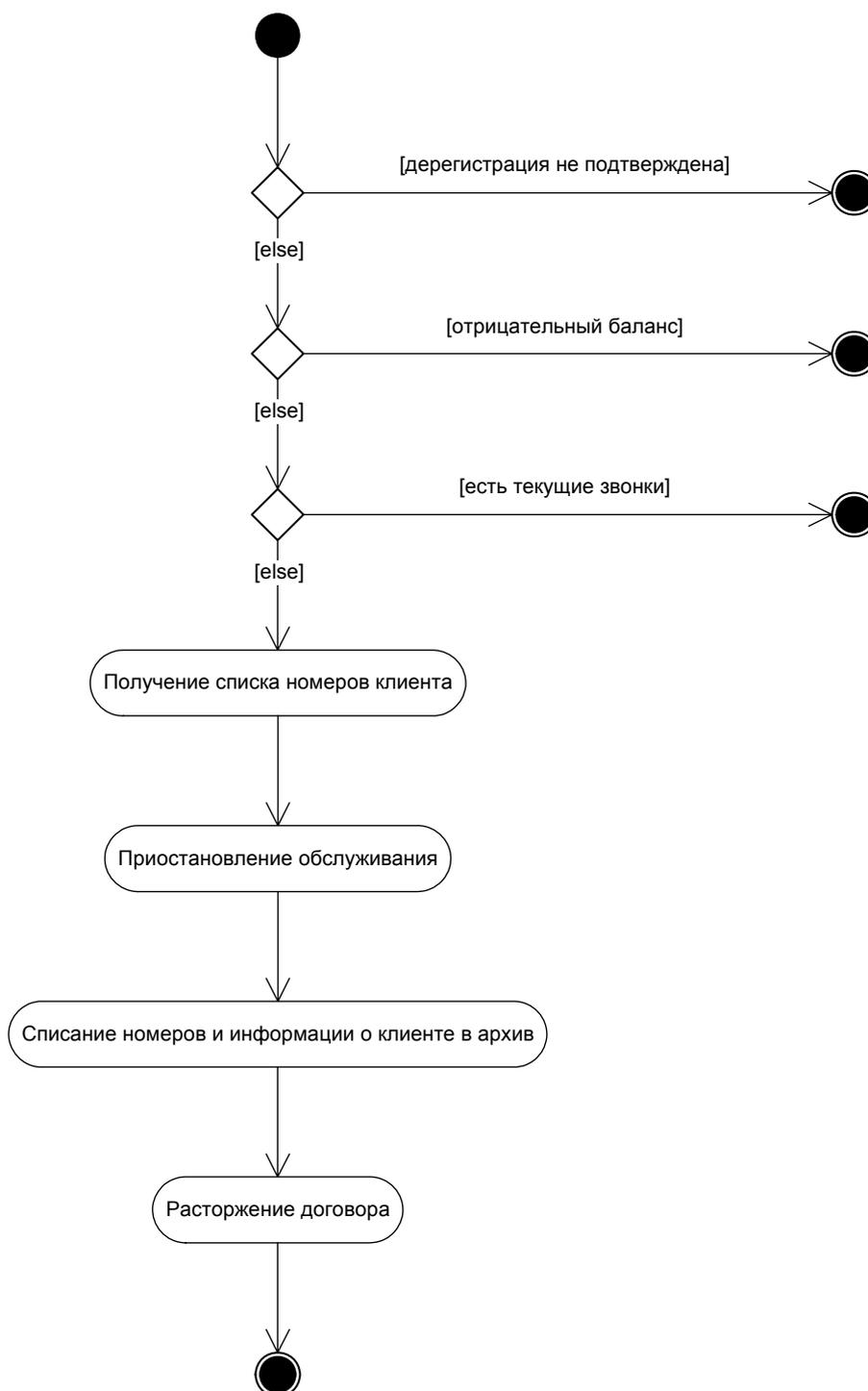


Рис. 3

Для выполнения этого варианта использования должны быть выполнены следующие условия:

- Клиент должен быть зарегистрирован
- Система не позволяет deregистрировать клиента, имеющего отрицательный баланс по счету
- У клиента не должно быть текущих звонков

При не выполнении этих условий, система должна генерировать соответствующие им исключения

4.3.3. Вариант использования “Занесение средств на счет”. Действия системы при реализации данного варианта использования изображены на диаграмме деятельности

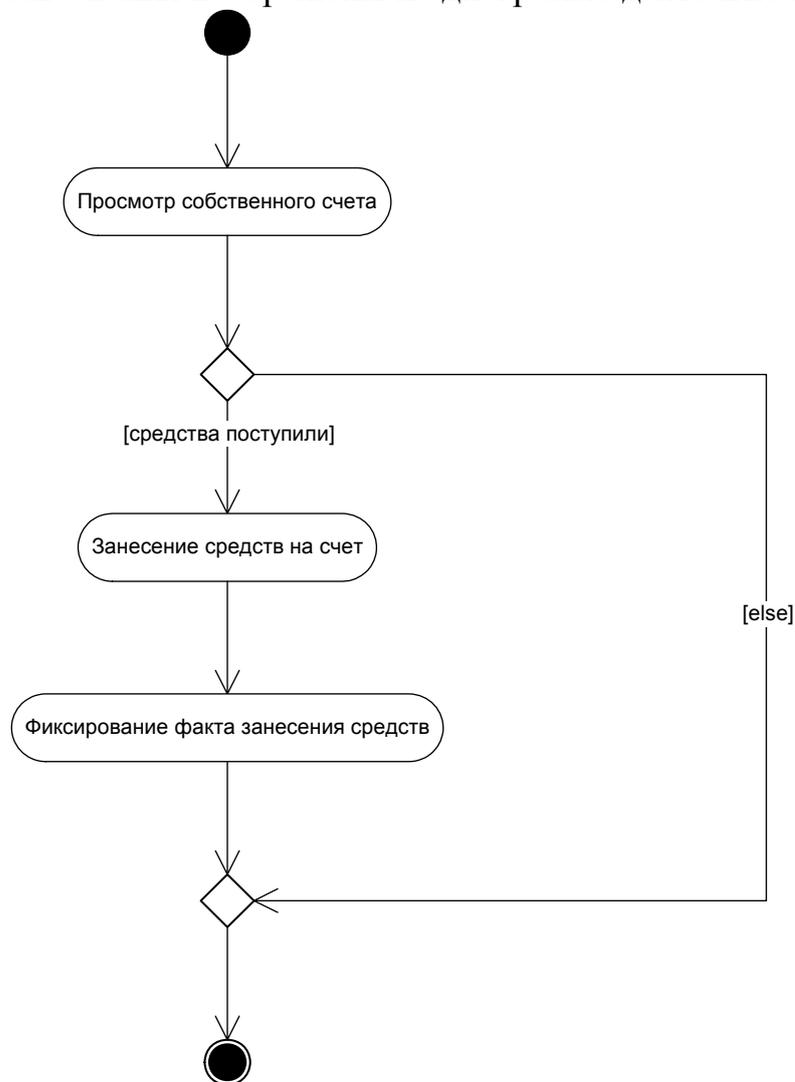


Рис. 4

4.3.4. Вариант использования “Регистрация тарифа” **Рис. 5**

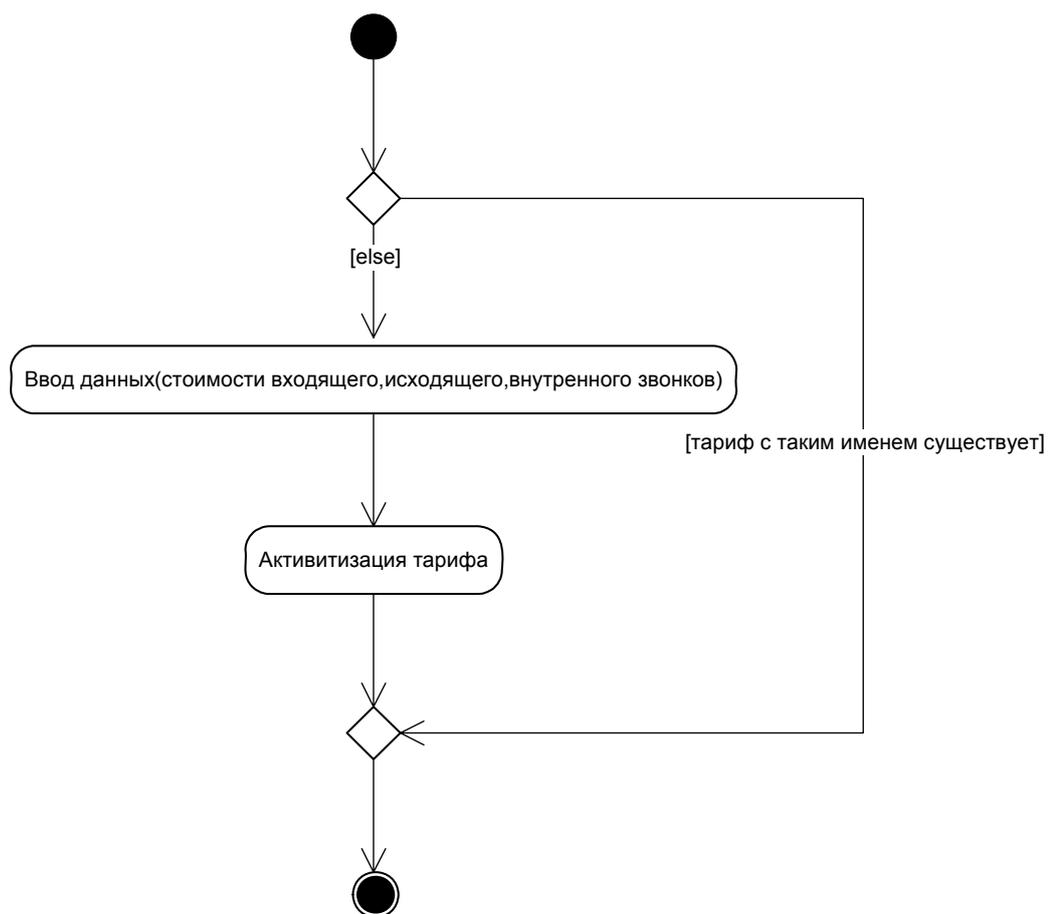


Рис. 5

Для выполнения этого варианта использования должны быть выполнены следующие условия:

- Тариф с таким именем должен отсутствовать

При не выполнении этих условий, система должна генерировать соответствующие им исключения

4.3.5. Вариант использования “Дерегистрация тарифа” Рис. 6

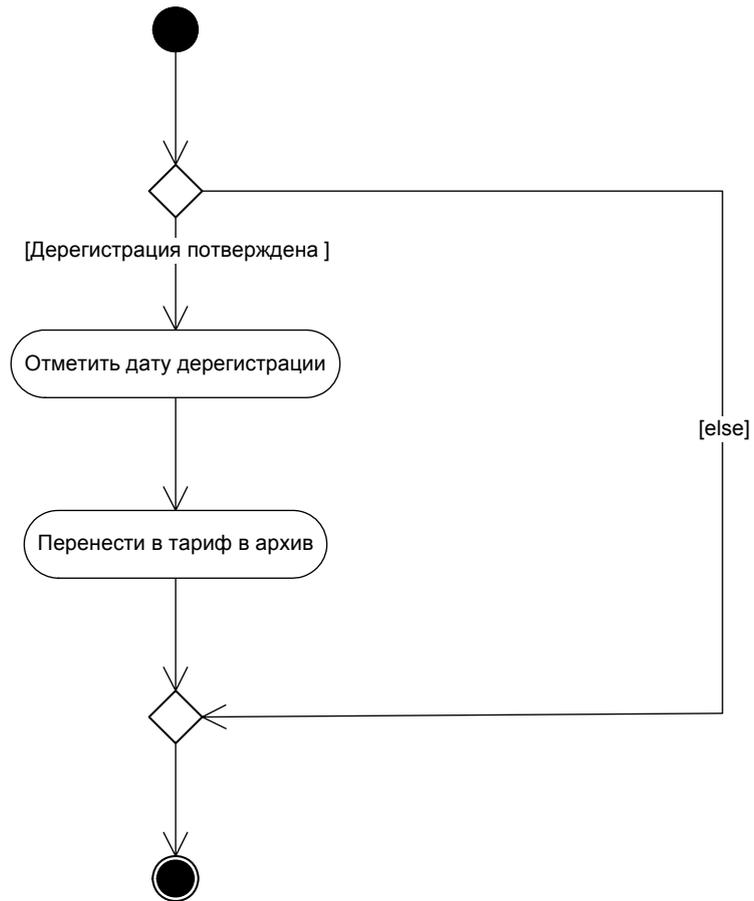


Рис. 6

4.3.6. Вариант использования “Ответ на звонок” Рис. 7

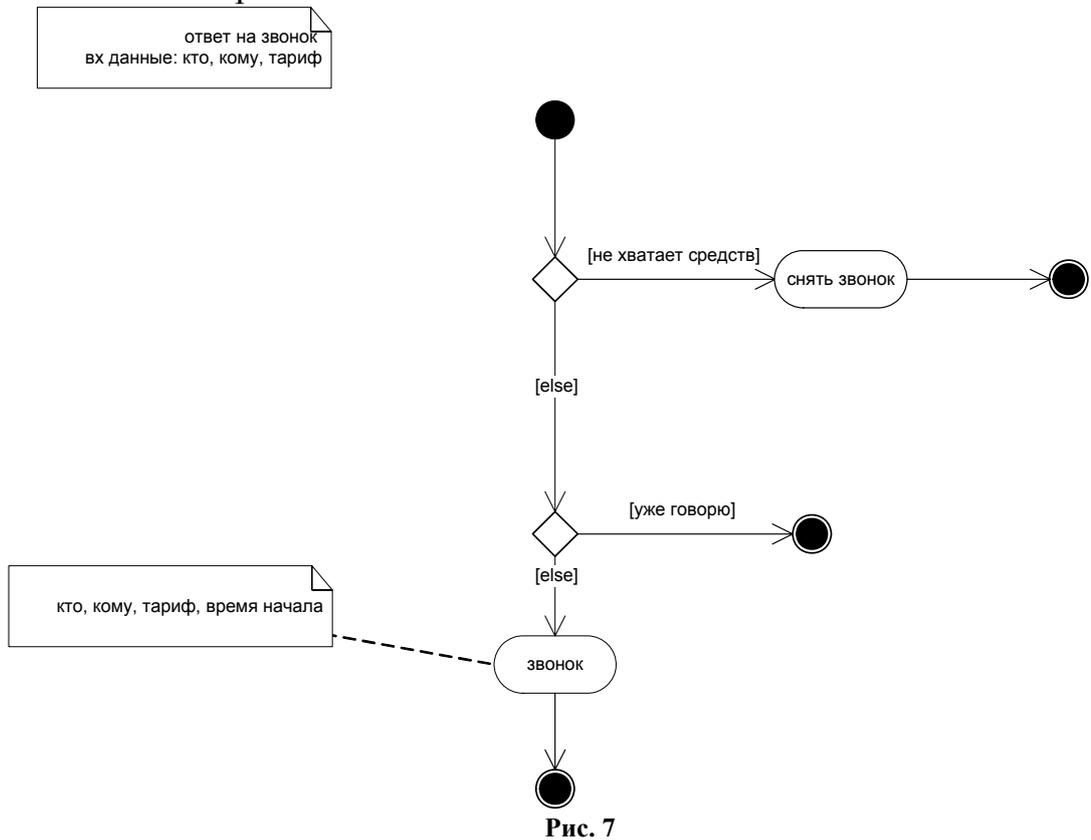


Рис. 7

Для выполнения этого варианта использования должны быть выполнены следующие условия:

- В системе не должно быть больше звонков, чем 3 – 4 тыс. звонков в минуту
- Пользователь должен иметь средства на счету для соединения по тарифу
- Пользователь должен находиться в зоне обслуживания

При не выполнении этих условий, система должна генерировать соответствующие им исключения

4.3.7. Вариант использования “Управление учетными записями” (*в разработке*)

4.3.8. Вариант использования “Изменение режима обслуживания” (*в разработке*)

4.3.9. Вариант использования “Формирование отчета” (*в разработке*)

4.3.10. Вариант использования “Окончание разговора” **Рис. 8**

окончание разговора

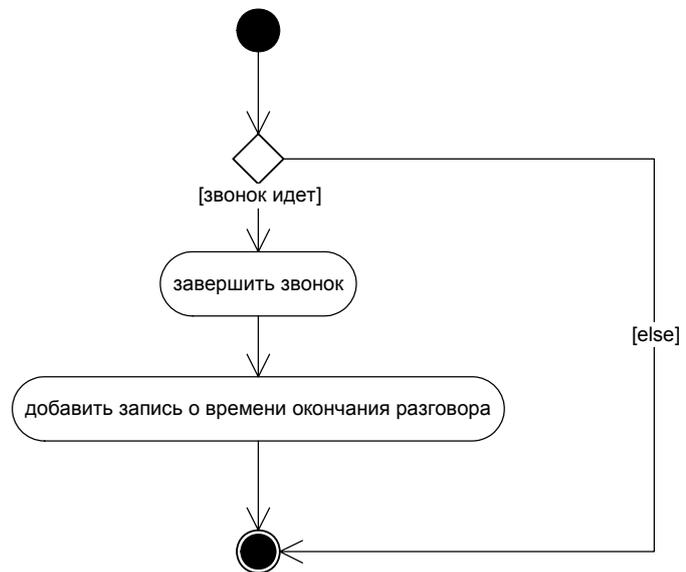


Рис. 8

4.3.11. Вариант использования “Звонок” **Рис. 9**

«Автоматизированные методы разработки архитектуры ПО.
Описание самостоятельной работы студентов (СРС)»

звонок
вх данные: кто, кому, тариф

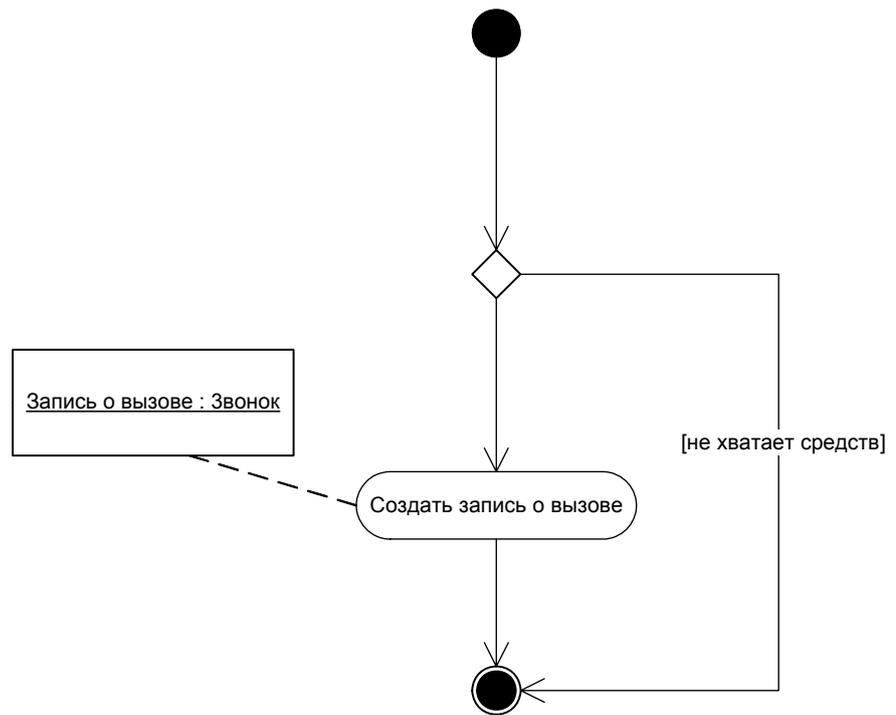


Рис. 9

4.4. Нефункциональные требования

4.4.1. Масштаб системы: 100-200 тыс. *клиентов*¹; пороговые значения 300 – 350 тыс. *клиентов*; 3 – 4 тыс. звонков в минуту.

4.4.2. Звонки происходят только внутри сети, т.е. между *клиентами* системы.

4.4.3. Занесенные данные о клиентах, тарифах и платежах хранятся вечно, и архивируются каждые 3 месяца

4.4.4. Требование к надежности. В системе должна быть функция архивирования данных, которая должна архивировать данные каждые три месяца с момента запуска. Система должна иметь возможность восстановления данных из указанного архива.

4.4.5. Требования к производительности: Подсистема учета звонков должна работать в режиме реального времени. Отчеты формируются автоматически не более 10 – 15 минут

4.4.6. Условия эксплуатации. Зависит от технической стороны проекта (Железо).

4.4.7. Требования к составу и параметрам технических средств. Техническое оборудование предоставляется заказчиком.

4.4.8. Требования к информационной и программной совместимости. Пользовательский интерфейс должен реализован на С#. База данных и функции обработки должны быть реализованы в СУБД MSSQL Server 2000.

4.4.9. Эксплуатационное назначение системы: Управление счетами, управление тарифами, управление учетными записями клиентов, ведение полной истории, тарификация и учет звонков в телефонной сети.

4.4.10. Требования к программной документации.

- Техническое задание пишется по ГОСТ 19.201-78 ЕСПД.
- Общее описание и требования к содержанию и оформлению пишутся по ГОСТ 19.502-78.
- Руководство пользователя пишется по ГОСТ 19.505-79

5. Особенности реализации

Система состоит из трех модулей:

- модуль администрирования, реализующий варианты использования “Регистрация нового тарифа”, “Дерегистрация нового тарифа”, “Управление учетными записями”, “Формирование отчета”

«Автоматизированные методы разработки архитектуры ПО.
Описание самостоятельной работы студентов (СРС)»

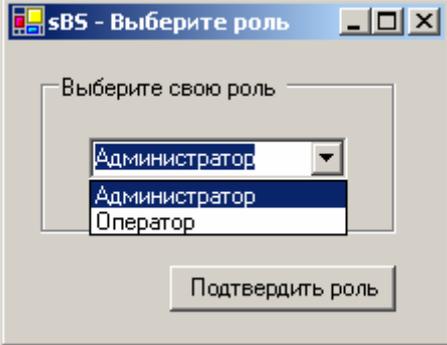
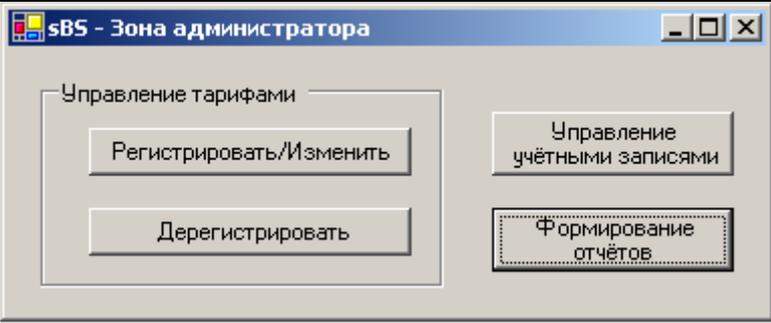
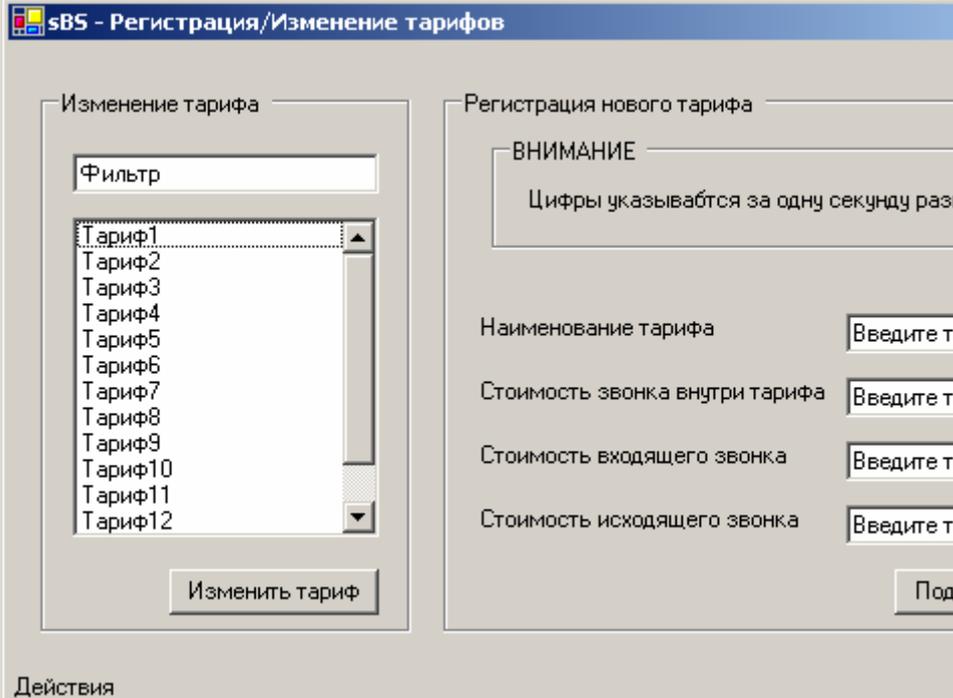
- модуль оператора, реализующий варианты использования “Регистрация нового клиента”, “Дерегистрация клиента”, “Занесение средств на счет”, “Изменение средств режима на обслуживания”, “Формирование отчета”
- модуль ядра

Для целей тестирования необходимо реализовать приложение для моделирования деятельности пользователя, реализующее варианты использования “Звонок”, “Ответ на звонок”, “Окончание разговора”

- Модуль администрирования – standalone приложение
- Модуль оператора – standalone приложение
- Модуль ядра – пакет функций написанные на Transact-Sql в среде MSSRQL SERVER 2000

6. Требования к GUI

6.1. Табл. 1 показывает окна доступные администратору

Название окна	Вид
<p>Меню выбора роли</p>	
<p>Зона администратора</p>	
<p>Меню регистрации тарифа</p>	

«Автоматизированные методы разработки архитектуры ПО.
Описание самостоятельной работы студентов (СРС)»

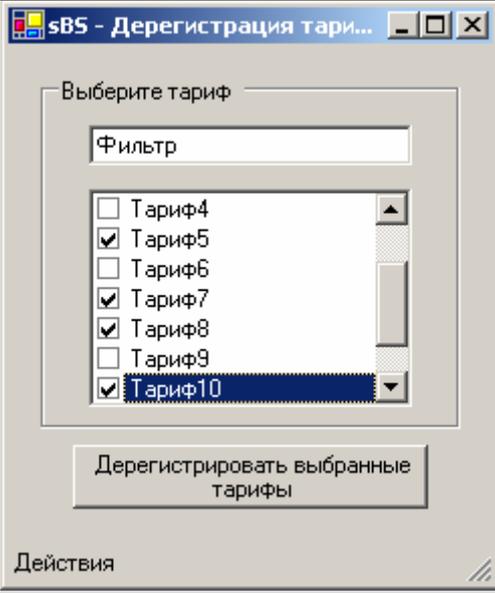
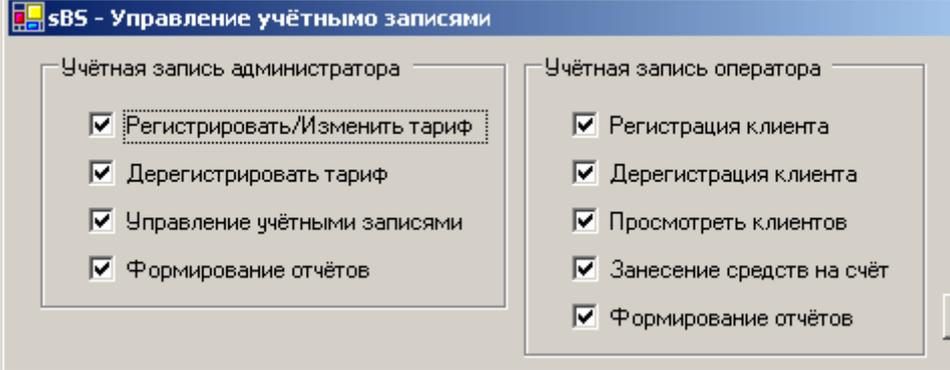
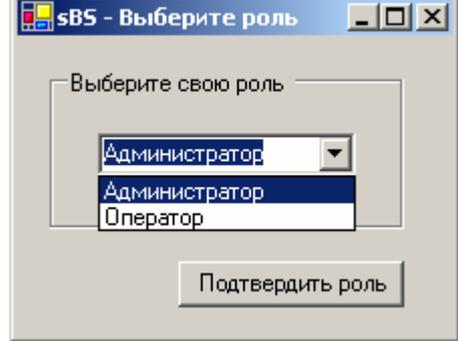
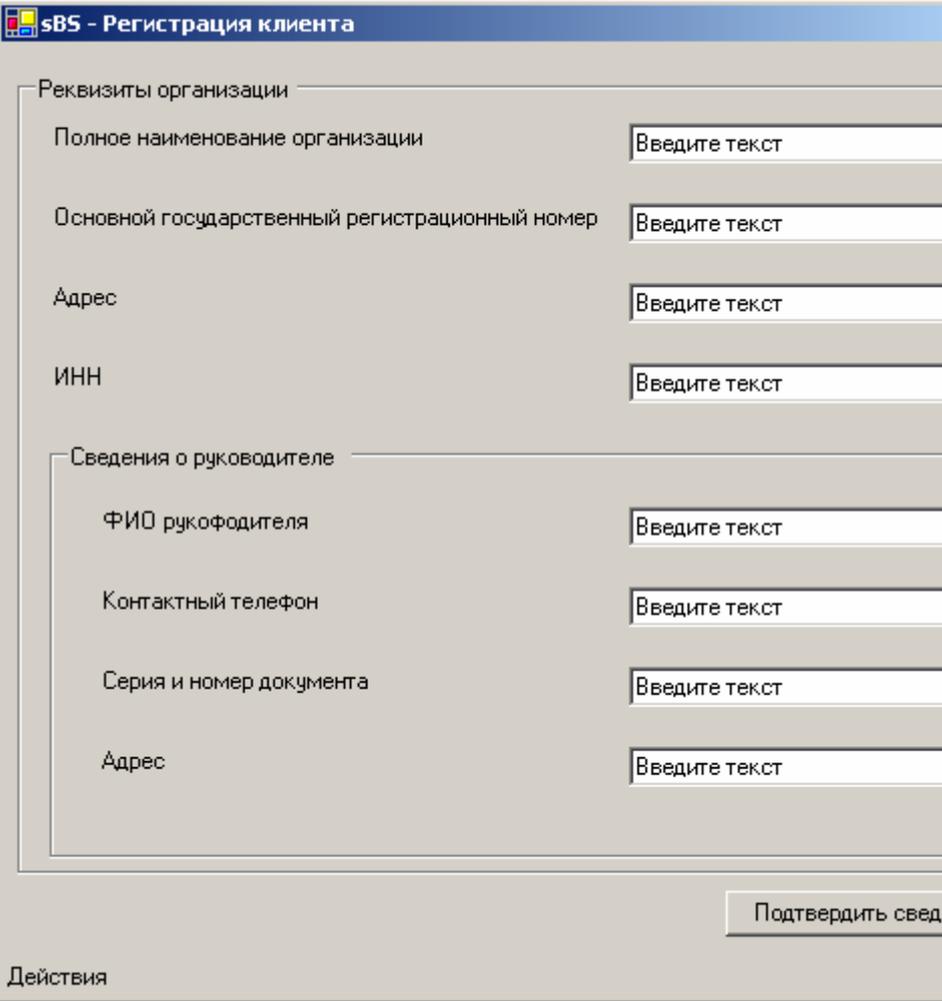
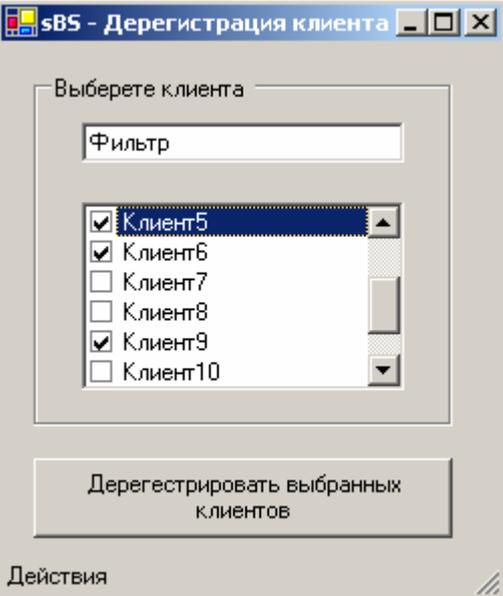
<p>Меню дерегистрац ии тарифа</p>	
<p>Управление учётными записями</p>	

Табл. 1

6.2. Табл. 2 показывает окна доступные оператору

<p>Название окна</p>	<p>Вид</p>
<p>Меню выбора роли</p>	

«Автоматизированные методы разработки архитектуры ПО.
Описание самостоятельной работы студентов (СРС)»

Регистрация клиента	 <p>Реквизиты организации</p> <p>Полное наименование организации <input type="text" value="Введите текст"/></p> <p>Основной государственный регистрационный номер <input type="text" value="Введите текст"/></p> <p>Адрес <input type="text" value="Введите текст"/></p> <p>ИНН <input type="text" value="Введите текст"/></p> <p>Сведения о руководителе</p> <p>ФИО руководителя <input type="text" value="Введите текст"/></p> <p>Контактный телефон <input type="text" value="Введите текст"/></p> <p>Серия и номер документа <input type="text" value="Введите текст"/></p> <p>Адрес <input type="text" value="Введите текст"/></p> <p align="right">Подтвердить сведения</p> <p>Действия</p>
Дерегистрация клиента	 <p>Выберете клиента</p> <p>Фильтр <input type="text"/></p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Клиент5<input checked="" type="checkbox"/> Клиент6<input type="checkbox"/> Клиент7<input type="checkbox"/> Клиент8<input checked="" type="checkbox"/> Клиент9<input type="checkbox"/> Клиент10 <p align="center">Дерегистрировать выбранных клиентов</p> <p>Действия</p>

«Автоматизированные методы разработки архитектуры ПО.
Описание самостоятельной работы студентов (СРС)»

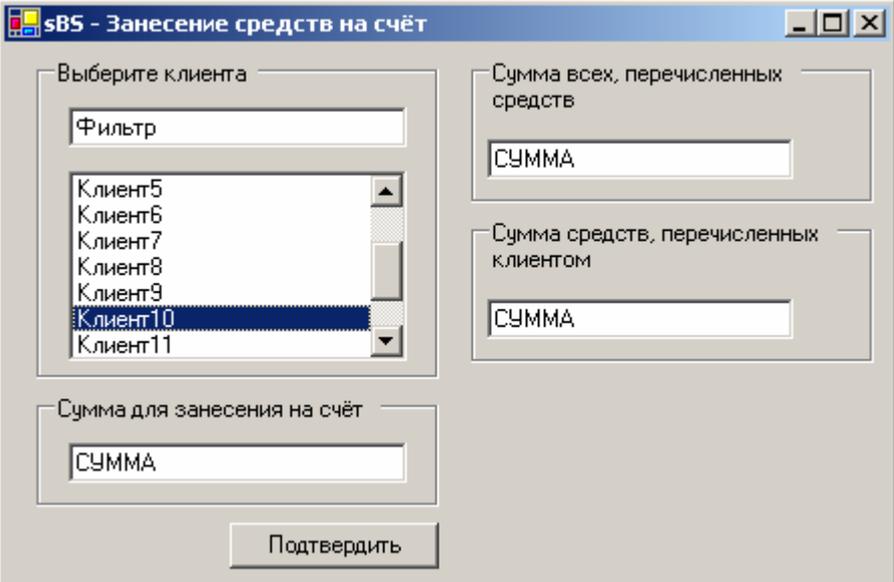
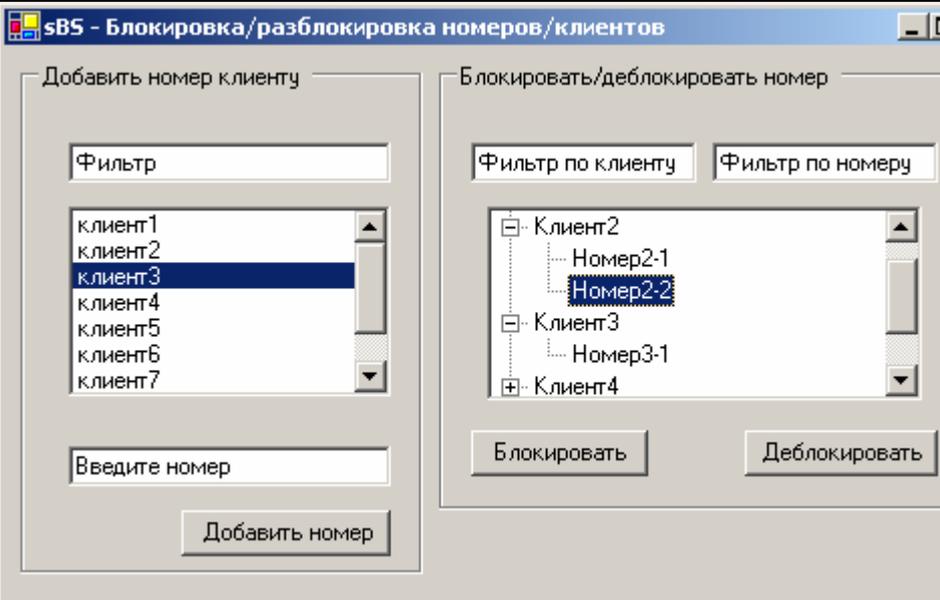
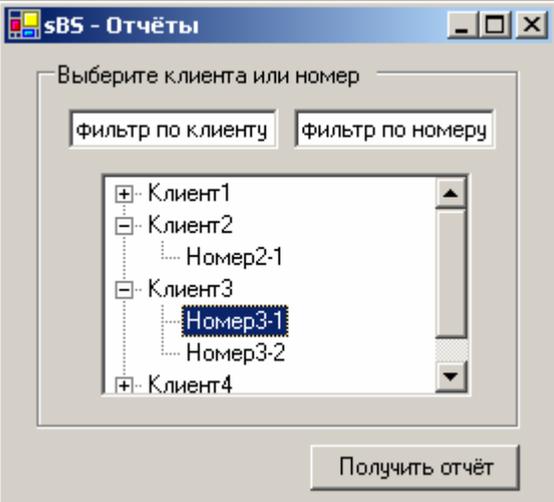
<p>Занесение средств на счет</p>	
<p>Меню блокировки и разблокировки клиентов</p>	
<p>Формирование отчета</p>	

Табл. 2

7. Стадии и этапы разработки

Стадия	Вход	Выход
Сбор и анализ требований	Протокол	Спецификация требований Одностраничное описание проекта
Проектирование системы	Спецификация требований	Документ по архитектуре проекта (Software Design Documentation)
Реализация	Проектная спецификация, график работ	Код
Тестирование	Спецификация требований, тестовые сценарии	Отчет о тестировании
Установка	Программная документация	Отчет об установке

Табл. 3

8. Глоссарий.

1. **Клиент** – юридическое лицо подписанное на пользование системой.
2. **Тариф** – набор правил оплаты клиентом его разговора.
3. **Тарификация** – непосредственное применение тарифа к звонку.
4. **Баланс** – количество средств на счету клиента.
5. **sBS** – smart Billing System
6. **Пользователь** – сотрудник клиента, имеющий номер
7. **Номер** – набор символов, идентифицирующий пользователя и имеющий следующий формат: максимальное число знаков – 8; первые два знака возможно латинские буквы.

Оглавление

Введение.....	2
Создание диаграммы Вариантов Исползования (Use Case)	5
Создание диаграммы Вариантов Исползования и действующих лиц.....	5
Этапы выполнения упражнения	5
Создание диаграмм взаимодействия.....	7
Создание диаграммы Последовательности ввода нового заказа после завершения первого этапа работы	7
Настройка программной среды	8
Создание диаграммы Последовательности	8
Добавление на диаграмму действующего лица и объектов	8
Добавление сообщений на диаграмму	8
Добавление на диаграмму дополнительных объектов.....	9
Назначение ответственностей объектам.....	10
Соотнесение объектов с классами.....	10
Соотнесение сообщений с операциями	11
Создание Кооперативной диаграммы	13
Добавление сообщений на диаграмму	13
Добавление дополнительных объектов на диаграмму.....	14
Добавление дополнительных объектов на диаграмму.....	15
Соотнесение объектов с классами - 2	16
Соотнесение объектов с классами – 3.....	16
Соотнесение сообщений с операциями - 2.....	17
Соотнесение сообщений с операциями - 3	17
Диаграмма Состояний для класса “Заказ”	19
Добавление начального и конечного состояний	19
Добавление суперсостояния	20
Добавление оставшихся состояний.....	20
Описание состояний	20
Добавление переходов.....	21
Описание переходов	21
Описание варианта исползовани.....	22
Образец Технического Задания для оформления самостоятельных работ студентов	27
по проекту	27
Версия 1.0.....	27
Содержание.....	28
Рис. 1.....	31

«Автоматизированные методы разработки архитектуры ПО.
Описание самостоятельной работы студентов (СРС)»

Рис. 2.....	35
Рис. 3.....	37
Рис. 4.....	38
Рис. 5.....	39
Рис. 6.....	40
Литература.....	52

Литература

[1] Уэнди Боггс, Майкл Боггс. UML и Rational Rose. Издательство: Лори, 2000 г., ISBN: 5-85582-091-2