МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА ИНФОРМАЦИОННО-СЕТЕВЫХ ТЕХНОЛОГИЙ

КУРСОВАЯ РАБОТА (ПРОЕКТ)   
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | Е.О. Пятлина |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ |
| ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ СРЕДСТВАМИ ЯЗЫКА UML |
| по дисциплине: ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4631 |  |  |  | С.А. Гришин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2018

**Введение.**

В данной работе приведен проект построения информационной системы. В качестве предметной области рассмотрена «Онлайн-бронирование туров». Проектирование производилось с помощью специализированного программного обеспечения Software Ideas Modeler – программного пакета для создания диаграмм UML. Приложение полностью совместимо с UML 2.2 и включает всевозможные типы диаграмм, используемых в этой системе.

**1. Описание основных функций Информационной системы «Онлайн-бронирование туров»:**

* Проверка наличия путёвок
* Работа администратора с базой данных
  + Добавление новых туров в базу данных
  + Обновление и редактирование информации о доступных турах
* Регистрация нового пользователя
  + Регистрация по электронной почте
  + Регистрация по номеру телефона
  + Подписаться на новостную рассылку при регистрации
* Поиск тура
  + Выбор страны
  + Выбор города
  + Выбор отеля
* Заказ путёвки
  + Заказ через Интернет
  + Заказ в отделении турагентства
  + Заказ по телефону
  + Подключение СМС-оповещений о состоянии заказа
* Покупка путёвки
  + Выбор способа оплаты покупки
  + Возможность использования скидки
* Выбор способа оплаты заказа
  + Наличными
  + Банковской картой
* Составление отчетности турфирмы
  + Получение отчётов о проданных турах
  + Получение отчётов о планируемых турах

**4. Разработка ПО информационной системы «Онлайн-бронирование туров».**

**4.1. Use-case диаграмма** (диаграмма вариантов использования, сценариев, прецедентов). Диаграммы позволяют наглядно представить ожидаемое поведение системы. Элементы, используемые на диаграмме:

* Сценарий. (Определяет один фрагмент поведения системы, без раскрытия внутреннего содержания)

Актер. (Внешняя по отношению к главной системе сущность, которая участвует в сценариях, является инициатором, источником или приемником информации, внутреннее содержание не рассматривается)  
В  созданной диаграмме использовались все возможные виды связей между  элементами  диаграммы:

* коммуникация (communication),
* включение (include),
* расширение (extend),
* обобщение (generalization).

Диаграммы Use Case по концепции напоминают классические DFD диаграммы, также применяемые для структурного анализа. Use Case также отображают границы исследуемой системы, её функциональность и определяют сущности и процессы, а также пользователей системы. При составлении этих диаграмм существует некоторая неопределённость. Она состоит в проблеме определения различий между сущностями и пользователями системы, а также её администраторами. Диаграммы Use Case чаще всего используются на этапе трансформации логической архитектуры системы в концептуальную модель, реализуемую при объектном подходе с определением событийной структуры управления будущим программным проектом.

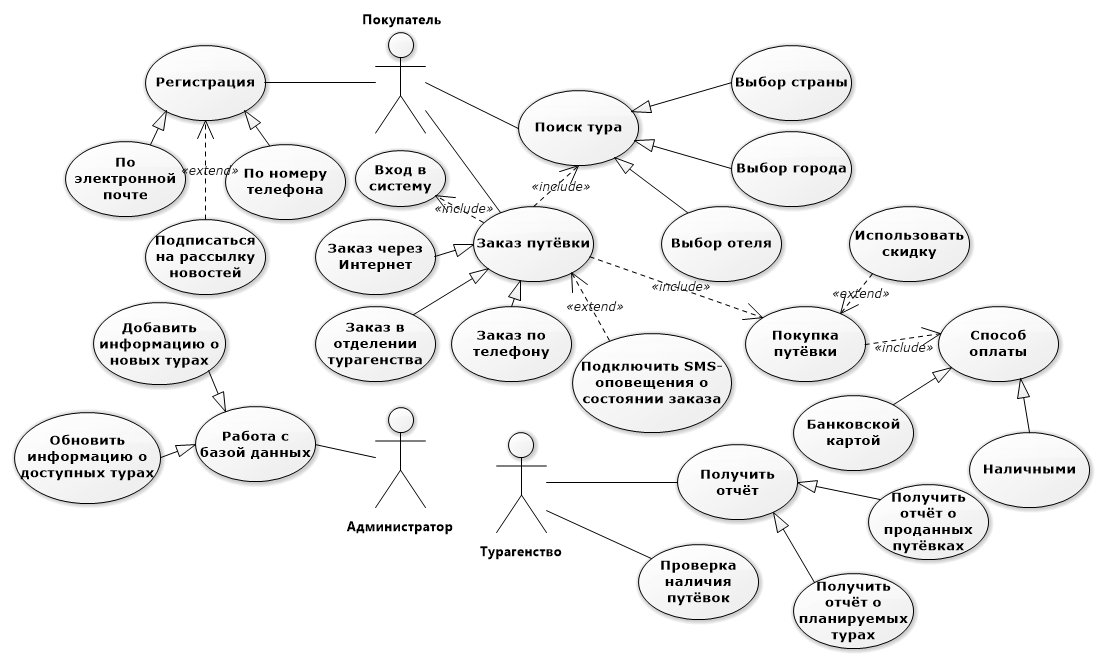
****

Рис.1 Общая USE-CASE диаграмма

**Расчет оценки диаграммы.   
**

где *Sobj*-оценка элемента на диаграмме, *Slink*- оценка связей, *Оbj*- кол-во объектов на диаграмме, *Tobj*–количество типов объектов, *Tlink*- количество типов связи.

**4.2. Диаграмма классов.**

[**Диаграмма классов**](http://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%BE%D0%B2) (Class diagram) — статическая структурная диаграмма, описывающая структуру системы, она демонстрирует классы системы, их атрибуты, методы и зависимости между классами. Существуют разные точки зрения на построение диаграмм классов в зависимости от целей их применения:

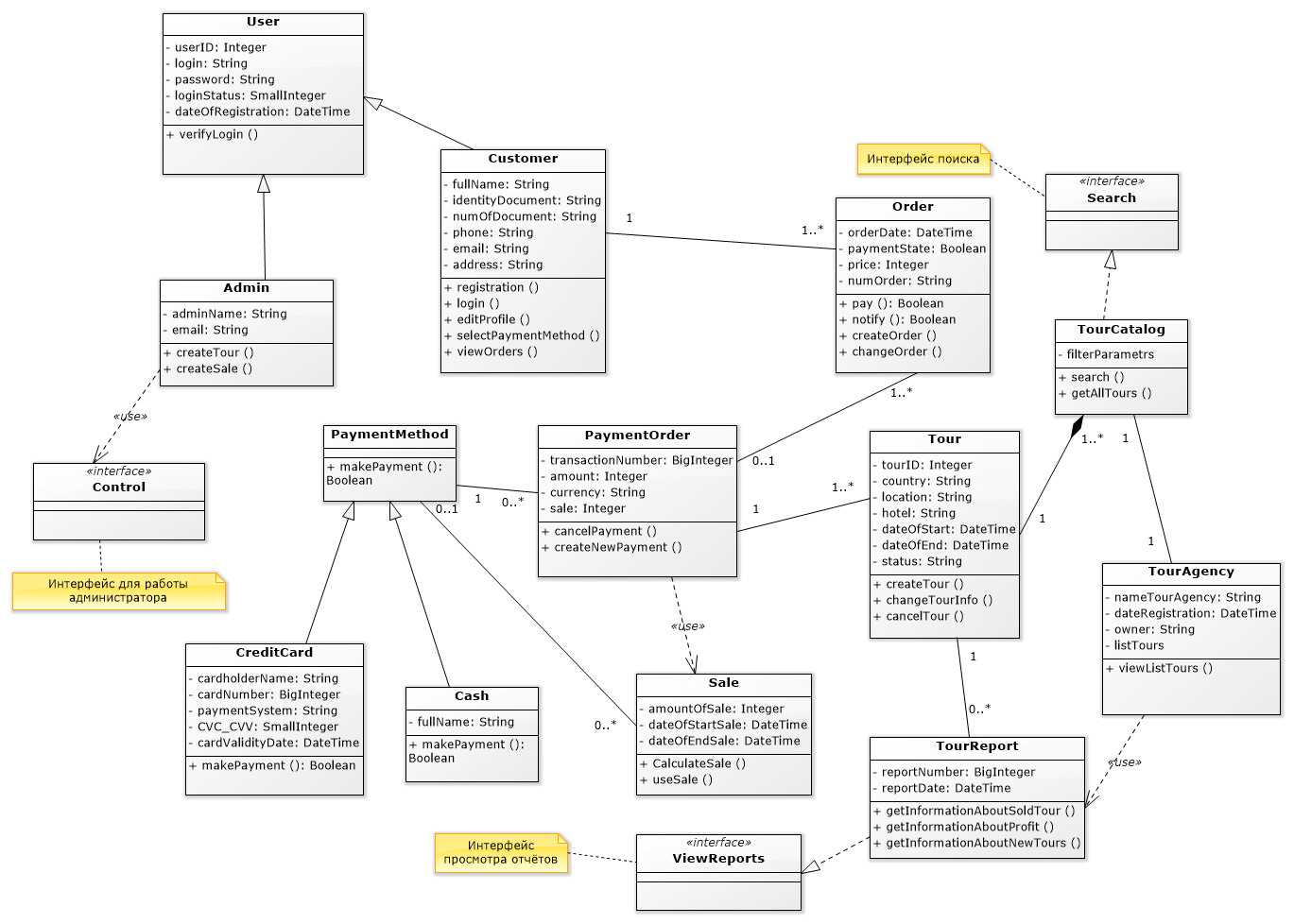
* концептуальная точка зрения — диаграмма классов описывает модель предметной области, в ней присутствуют только классы прикладных объектов;
* точка зрения спецификации — диаграмма классов применяется при проектировании информационных систем;
* точка зрения реализации — диаграмма классов содержит классы, используемые непосредственно в программном коде (при использовании объектно-ориентированных языков программирования).  
  ****

Рис. 2 Диаграмма классов.

**Расчет оценки диаграммы.**

где *Sobj*-оценка элемента на диаграмме, *Slink* - оценка связей, *Оbj* - кол-во объектов на диаграмме, *Tobj*–количество типов объектов, *Tlink*- количество типов связи.

где *Op*- количество операций, *Atr*- количество атрибутов.

**Расчет оценки диаграммы классов с атрибутами и операциями:**

**4.3. Диаграммы последовательностей.**

**Диаграммы последовательностей *(***[*англ.*](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA)*Sequence diagram)* — [диаграмма](http://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), на которой показаны взаимодействия объектов, упорядоченные по времени их проявления. Основными элементами диаграммы последовательностей являются обозначения объектов (прямоугольники), вертикальные линии ([*англ.*](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *lifeline*), отображающие течение времени при деятельности объекта, и стрелки, показывающие выполнение действий объектами.

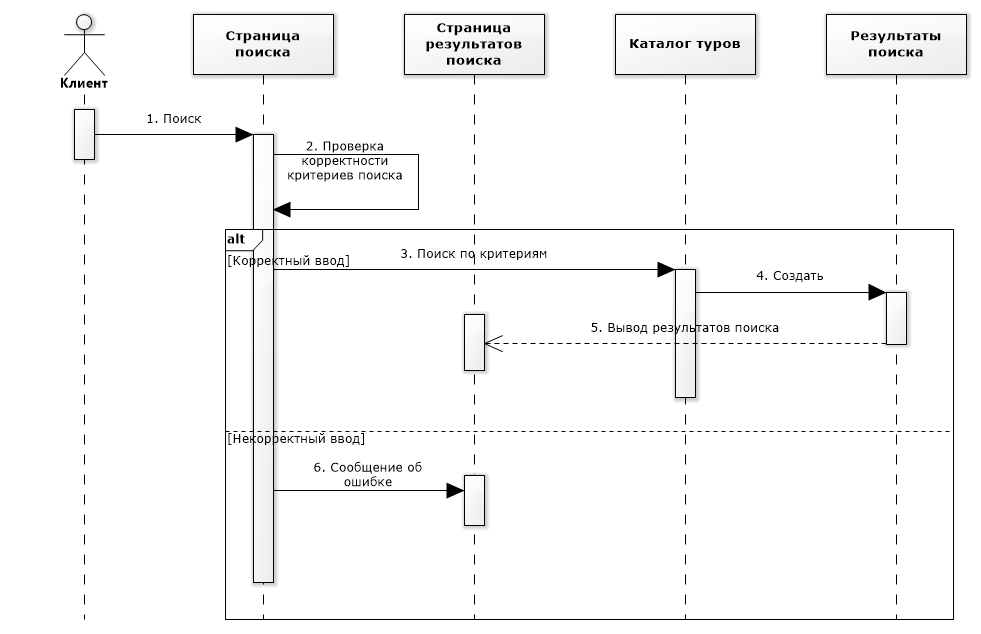


Рис.3 Диаграмма последовательностей для сценария «Поиск тура».

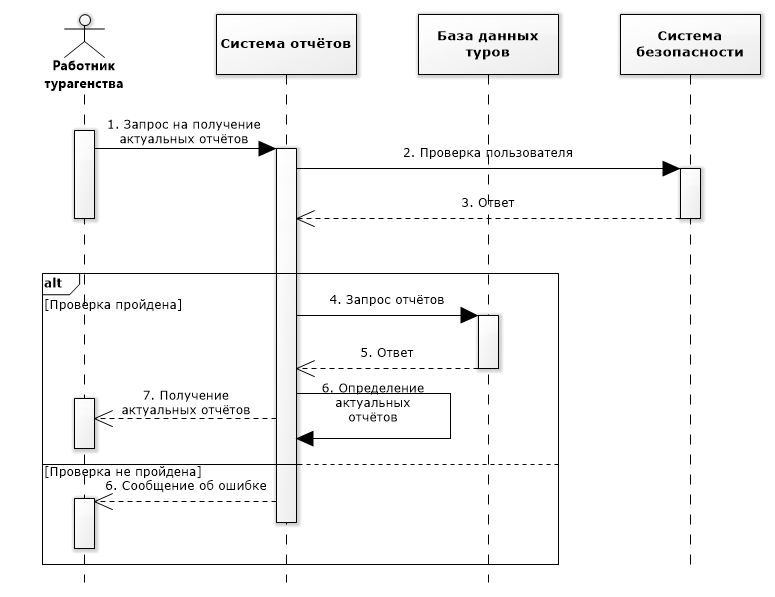
****

Рис.4 Диаграмма последовательностей для сценария «Работа с отчётами турфирмы».

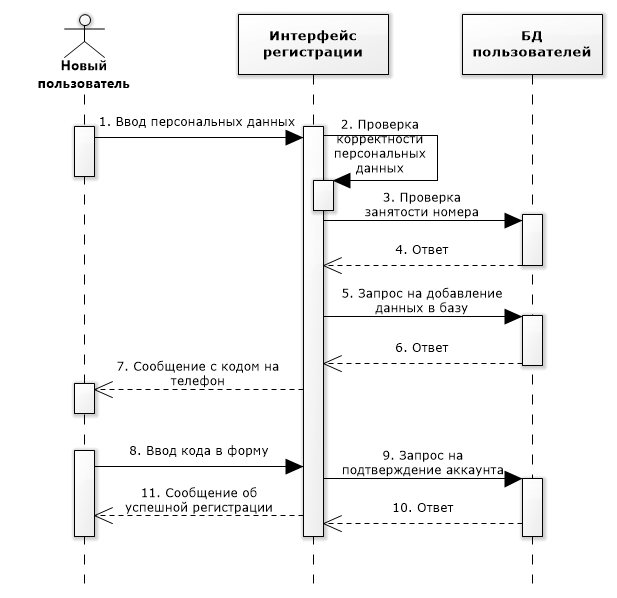
****

Рис.5 Диаграмма последовательностей для сценария «Регистрация пользователя по e-mail».

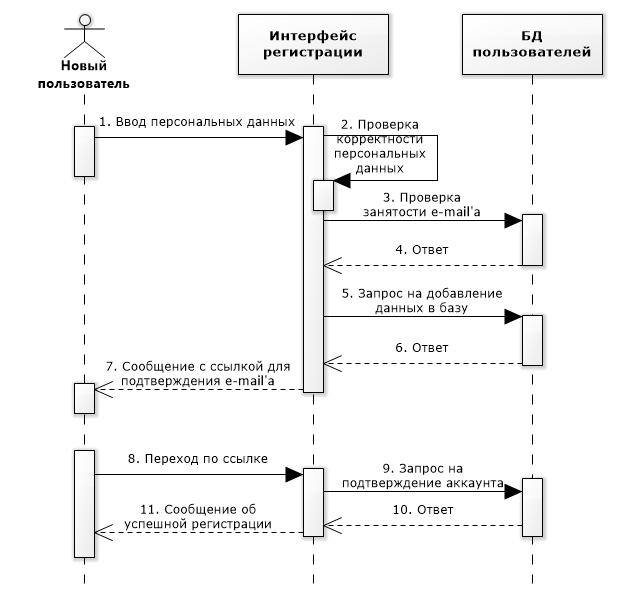
****

Рис.6 Диаграмма последовательностей для сценария «Регистрация пользователя по номеру телефона».

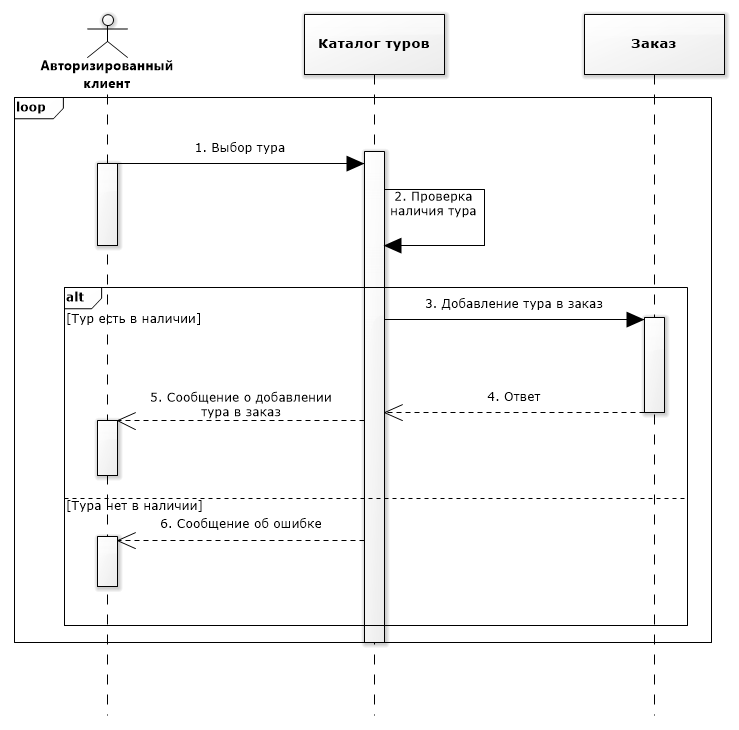
****

Рис.7 Диаграмма последовательностей для сценария «Заказ тура».

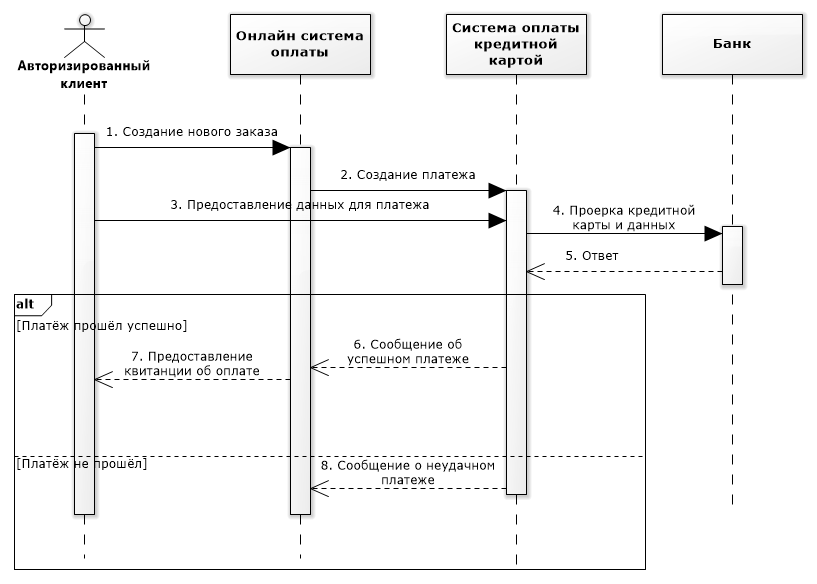
****

Рис.8 Диаграмма последовательностей для сценария «Оплата заказа».

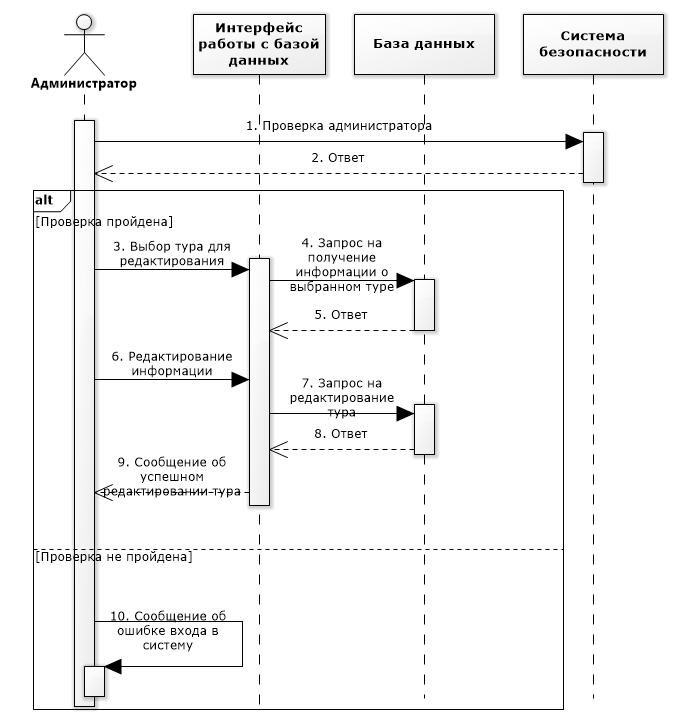
****

Рис.9 Диаграмма последовательностей для сценария «Работа с базой данных».

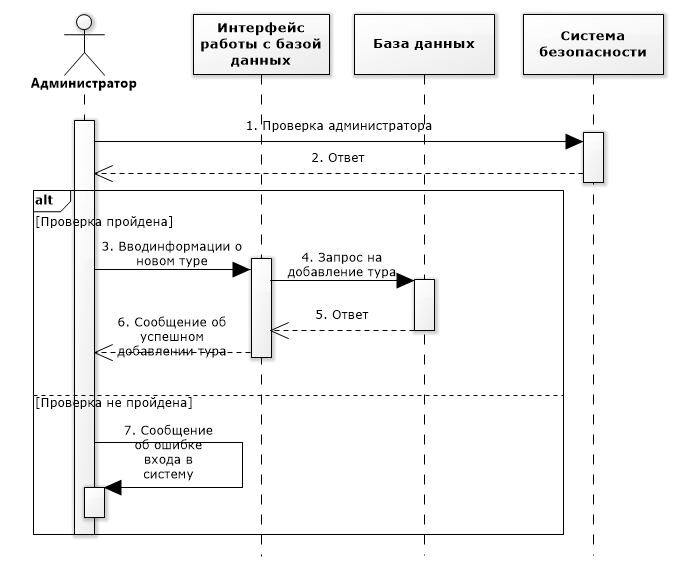
****

Рис.10 Диаграмма последовательностей для альтернативного сценария «Работа с базой данных».

**4.4. Диаграммы состояний (Statechar diagram)**

Диаграмма состояний (Statechar diagram) определяют все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате наступления некоторых событий.

Существует много форм диаграмм состояний, незначительно отличающихся друг от друга семантикой.

На диаграмме имеются два специальных состояния – начальное (start) и конечное(stop). Начальное состояние выделено черной точкой, оно соответствует состоянию объекта, когда он только что был создан. Конечное состояние обозначается черной точкой в белом кружке, оно соответствует состоянию объекта непосредственно перед его уничтожением. На диаграмме состояний может быть одно и только одно начальное состояние. В то же время, может быть столько конечных состояний, сколько вам нужно, или их может не быть вообще. Когда объект находится в каком-то конкретном состоянии, могут выполняться различные процессы. Процессы, происходящие, когда объект находится в определенном состоянии, называются действиями (actions).



Рис 11. Диаграмма состояний для класса «User».

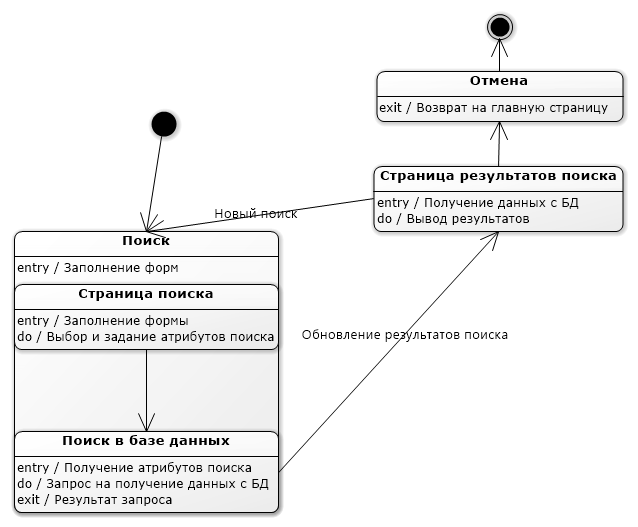


Рис 12. Диаграмма состояний для класса «Search».

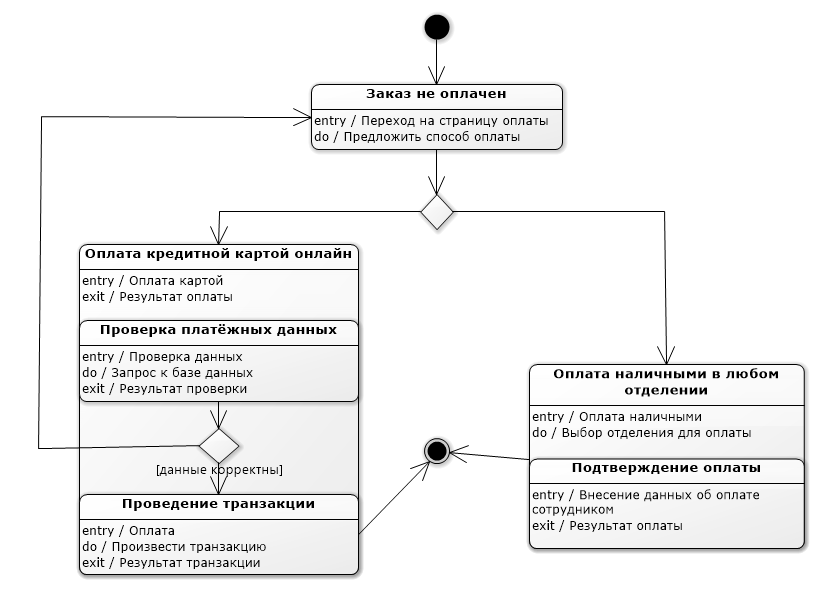


Рис 13. Диаграмма состояний для класса «PaymentOrder».

Рис 14. Диаграмма состояний для класса «CreditCard».

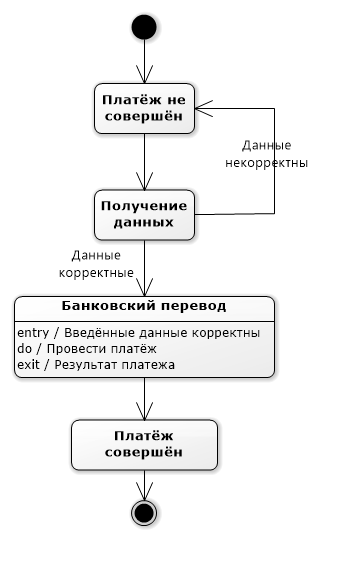


Рис 15. Диаграмма состояний для класса «Cash».

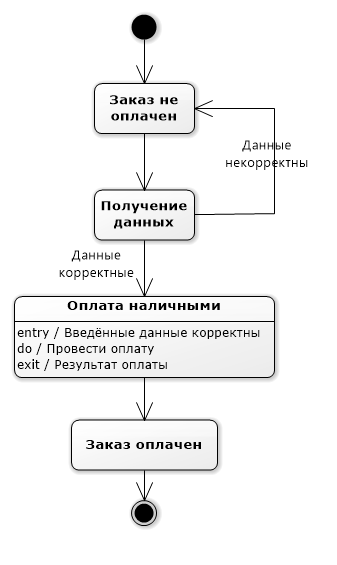


Рис 16. Диаграмма состояний для класса «Sale».



Рис 17. Диаграмма состояний для класса «Tour».

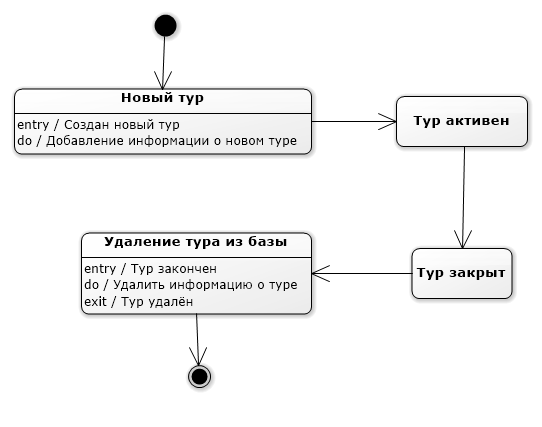
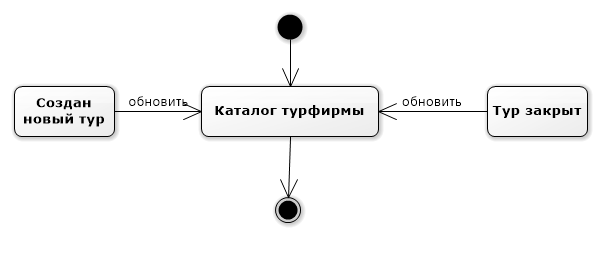
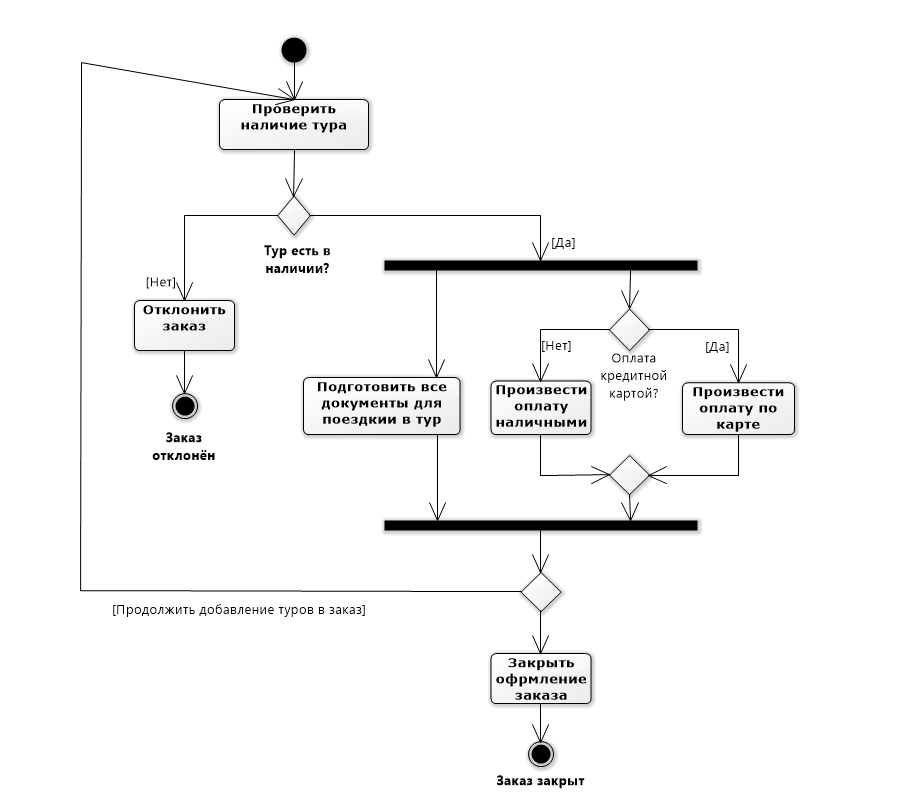


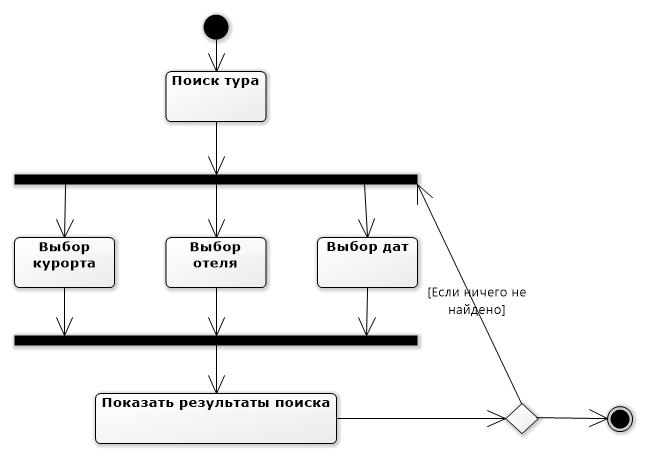
Рис 18. Диаграмма состояний для класса «TourCatalog».

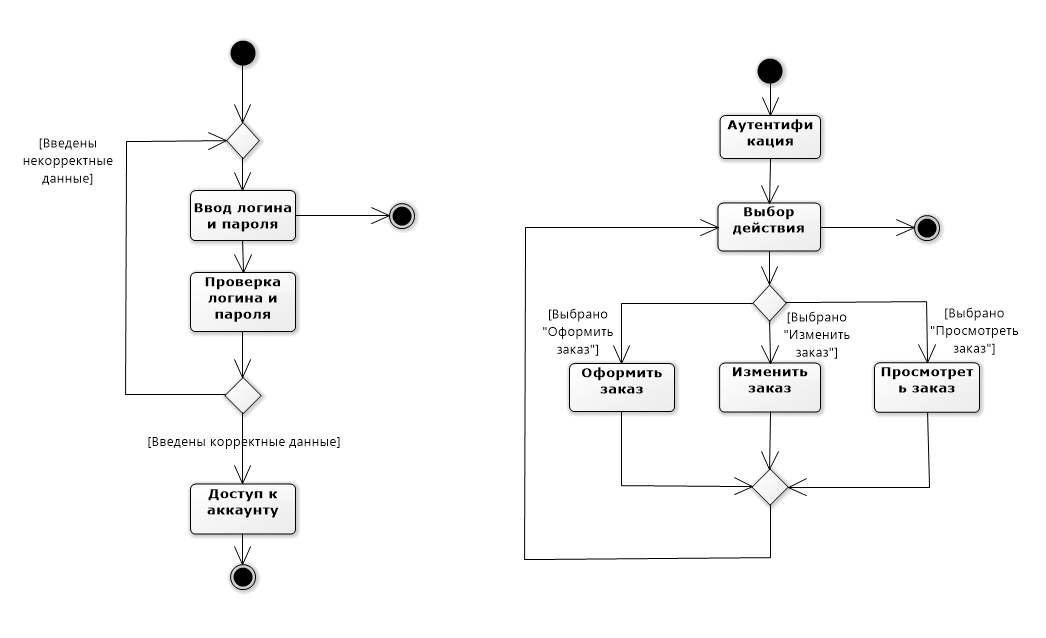


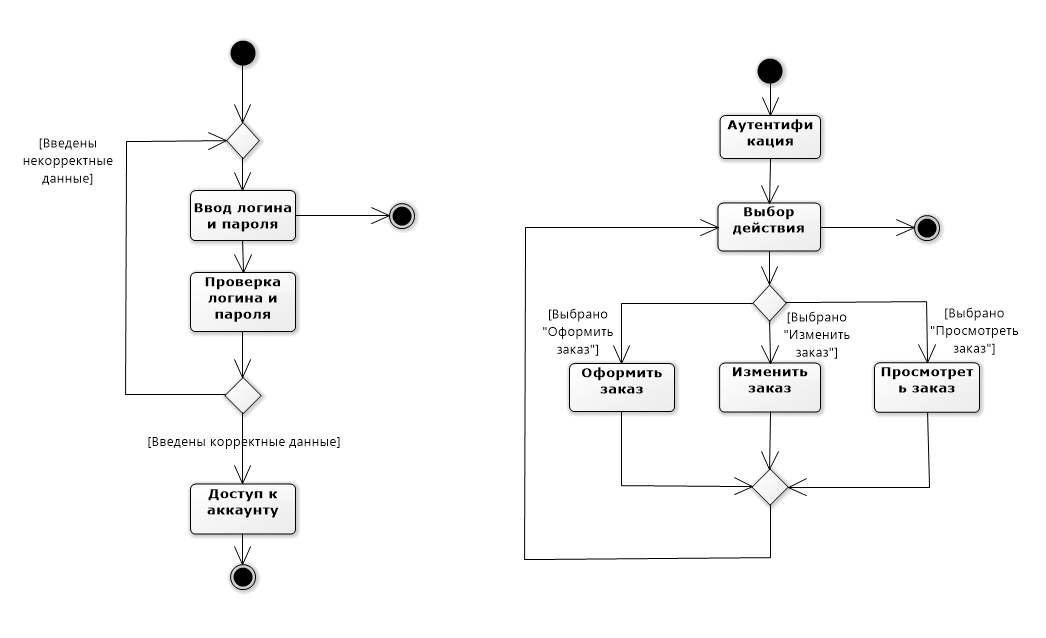
**4.5 Диаграммы видов деятельности(Activity diagram)**

Диаграммы видов деятельности(Activity diagram)-диаграмма, на которой показано разложение некоторой деятельности на ее составные части. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчиненных элементов - вложенных видов деятельности и отдельных действий, соединенных между собой потоками, которые идут от выходов одного узла ко входам другого.

  
Рис 19. Диаграмма деятельности для метода «CreateOrder» класса «Order».

  
Рис 20. Диаграмма деятельности для метода «Search» класса «TourCatalog».

  
Рис 21. Диаграмма деятельности для метода «Login» класса «Customer».

  
Рис 22. Диаграмма деятельности для метода «ChangeOrder» класса «Order».

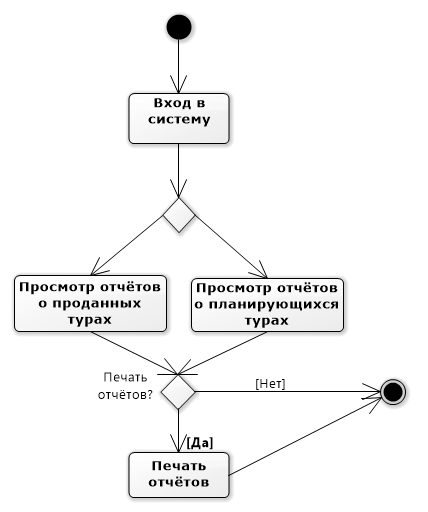
  
Рис 23. Диаграмма деятельности для метода «GetInformationAboutTours» класса «TourReport».

Рис 24. Диаграмма деятельности для метода «MakePayment» класса «PaymentMethod».

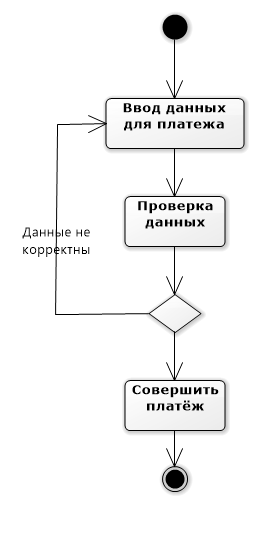


Рис 25. Диаграмма деятельности для метода «ViewListTours» класса «TourAgency».



Рис 26. Диаграмма деятельности для метода «CalculateSale» класса «Sale».

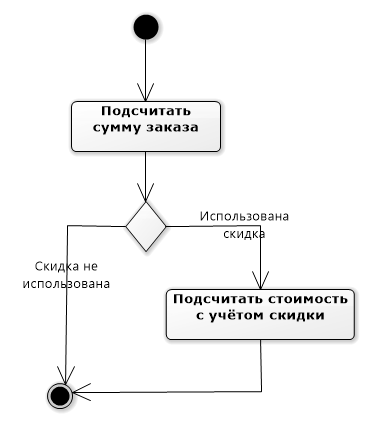
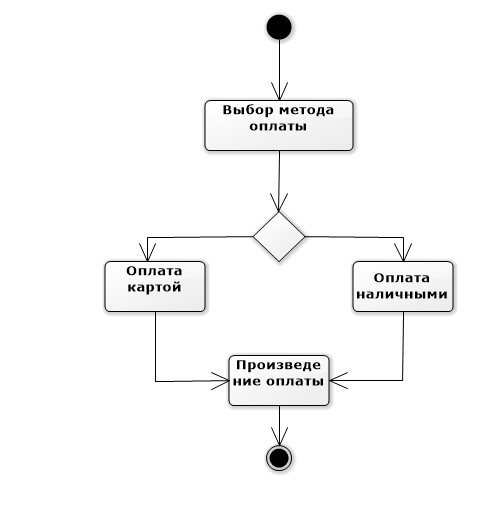
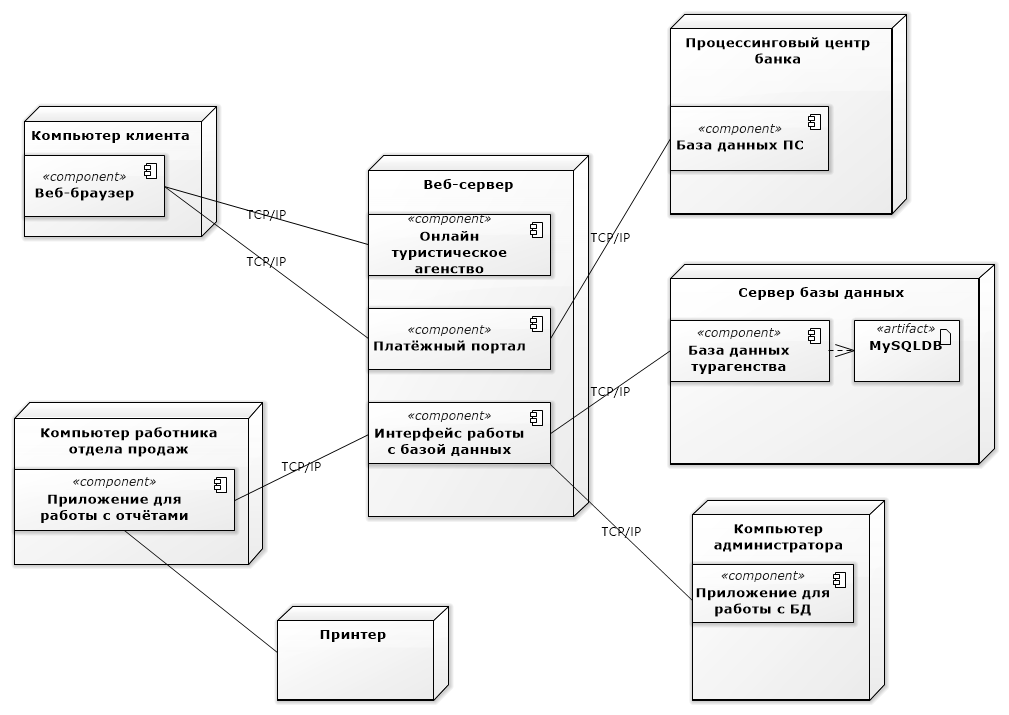


Рис 27. Диаграмма деятельности для метода «SelectPaymentMethod» класса «Customer».



**4.6.Диаграмма размещений (Диаграмма развертывания).**

Диаграмма развертывания (Deployment diagram)- предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения. Диаграмма развертывания отражает физические взаимосвязи между программными и аппаратными компонентами разрабатываемой системы. Каждый узел на диаграмме развертывания представляет собой некоторый тип вычислительного устройства - в большинстве случаев самостоятельную часть аппаратуры.



**ЗАКЛЮЧЕНИЕ**

В данной курсовой работе спроектирована информационная система «Онлайн-бронирование туров» в нотации UML с использованием Сase-средства Ideas Software Modeler. Система описана практически со всех возможных точек зрения, рассмотрены разные аспекты поведения системы, диаграммы сравнительно просты для чтения, методы описания результатов анализа и проектирования семантически близки к методам программирования на современных ОО-языках.

Данная работа дает возможность организовать качественное функционирование описанной информационной системы, позволяет автоматизировать процессы, что сэкономит и время, и средства.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Иванова Г.С. Технология программирования: учебник/Г.С. Иванова – М.:КНОРУС, 2011.-336с.
2. Леоненков А.В. Самоучитель UML: монография/ А. Леоненков. - СПб. и др.: BHV - Санкт-Петербург, 2010 - 298 с.
3. Буч Г., Якобсон А., Рамбо Дж. Язык UML. Руководство пользователя / Пер. с англ. — ДМК Пресс , 2008 г.
4. Ларман К. Применение UML 2.0 и шаблонов проектирования, Вильямс, 2009 г.
5. Фаулер М., Скотт К. UML. Основы, Символ-Плюс, 2006 г.