Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Тверской государственный технический университет»

(ФГБОУВПО «ТвГТУ»)

Кафедра "Программное обеспечение".

Курсовая работа по дисциплине

"Унифицированный язык UML".

ТЕМА:

Информатизация мониторинга ремонта ЖД вагонов в депо.

Выполнил:

Иванов Р.В

Студент второго курса

Факультета информационных технологий, группы ПИН 17.06,

Проверил:

Котлинский С.В.

Тверь

2019 г.

Оглавление

Оглавление

[Постановка задачи 2](#_Toc8590417)

[Цель 2](#_Toc8590418)

[Реализация 2](#_Toc8590419)

[Диаграмма вариантов использования 3](#_Toc8590420)

[Диаграмма последовательности 4](#_Toc8590421)

[Диаграмма кооперации 8](#_Toc8590422)

[Диаграмма классов 10](#_Toc8590423)

[Диаграмма состояний 12](#_Toc8590424)

[Диаграмма компонентов 13](#_Toc8590425)

[Диаграмма размещения 13](#_Toc8590426)

[Заключение 14](#_Toc8590427)

[Использованная литература 14](#_Toc8590428)

# Постановка задачи

Построить модель, проектируемой информационной системы на основе унифицированного языка моделирования UML (с использованием Rational Rose) с уровнем сложности, согласованным с руководителем работы.

# Цель

Создать систему для информатизация мониторинга ремонта ЖД вагонов в депо

# Реализация

**Общая картина ремонта:**

Клиент решает сдать свой вагон на ремонт в депо. Депо дает ответ клиенту, возьмется он ли ремонтировать вагон или нет. Если депо берется за ремонт, происходят следующие действия: вагон отправляется на мойку и разборку. Разборка в свою очередь происходит в два этапа: вагон разбираю на кузов, который в последствии отправится на ремонт и детали. Детали, разобранного вагона, сначала подвергаются очистки, затем разбираются на более мелкие детали, далее эти детали подвергаются контроль качества. Детали, прошедшие контроль качества отправляются на склад «Годные детали», а детали которые не прошли контроль качества на склад «Детали подлежащие ремонту» и «Негодные детали». Далее мы ремонтируем детали со склада «Детали подлежащие ремонту». Следующим шагом мы начинаем комплектование на сборку уже непосредственно деталей для самого вагон из сборочных единиц с которыми мы занимались ранее. Собранные детали мы тестируем, красим и отправляем на склад «Склад сборочных единиц». Затем мы приступаем к общий сборке вагона(кузов + детали).Собрав вагон мы испытываем его и красим, после чего отправляем клиенту.

# Диаграмма вариантов использования

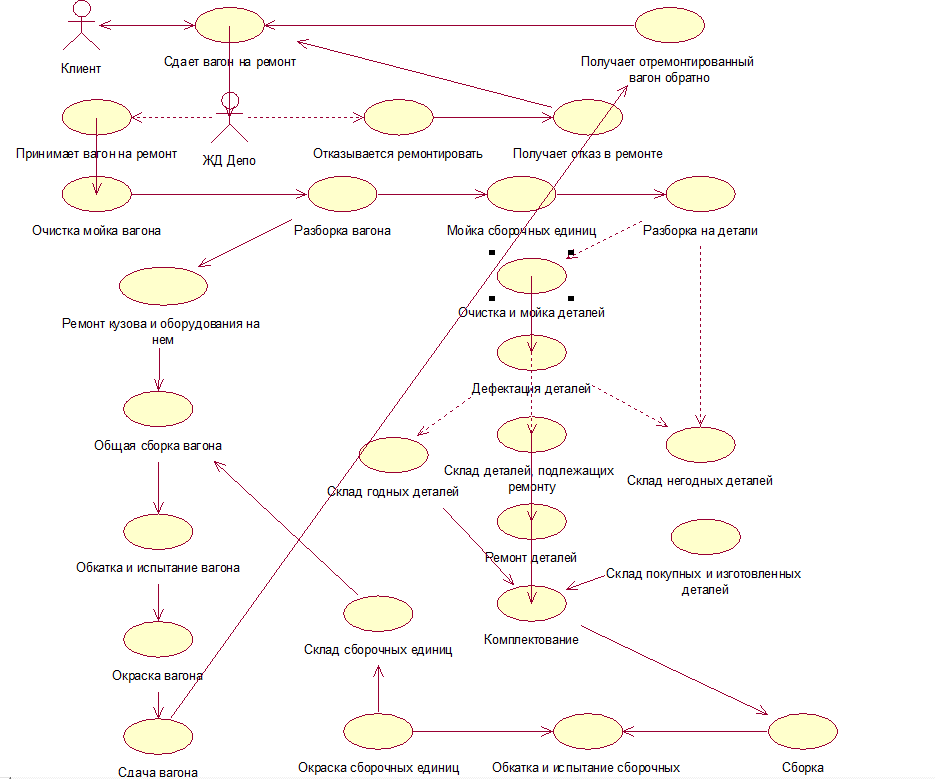


Рис. 1

**Диаграмма вариантов использования** (Use case diagram) – это графическое представление всех или части действующих лиц, вариантов использования и их взаимодействий в системе.   
  
В каждой системе есть главная диаграмма вариантов использования, которая отображает границы системы (действующие лица) и основное функциональное поведение системы (варианты использования). Другие диаграммы использования могут создаваться при необходимости.

**Прецеденты** – это механизм упрощения этапа формулировки требований для всех заинтересованных лиц. По существу, это рассказы об использовании системы в процессе решения поставленных задач.   
  
Основная идея состоит в исследовании и формулировке функциональных требований путем написания историй «из жизни системы». Эти истории помогают сформулировать различные задачи и представляют собой сценарии использования системы. На первый взгляд, описать прецеденты несложно, хотя зачастую трудно определить, что требуется от системы и описать это на нужном уровне детализации.  
  
Сила механизма прецедентов состоит в возможности масштабировать уровень сложности и формальности описания в зависимости от реальных потребностей.

**Действующее лицо** (Actor) – это сущность, обладающая поведением. Это роль, которую пользователь играет по отношению к системе. Действующие лица представляют собой роли, а не конкретных людей или наименования работ. К числу действующих лиц могут относиться и сама рассматриваемая система, если она вызывает службы других систем.

На диаграмме вариантов использования курсовой работы (рис. 1) в качестве актёров выбраны следующие сущности: клиент, жд депо.

Клиент – лицо, заказавшее починку вагона.

ЖД Депо – лицо, непосредственно занимающееся починкой вагона.

# Диаграмма последовательности

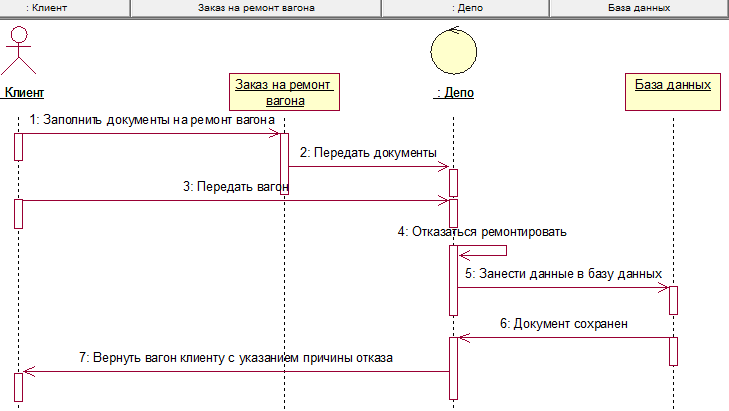
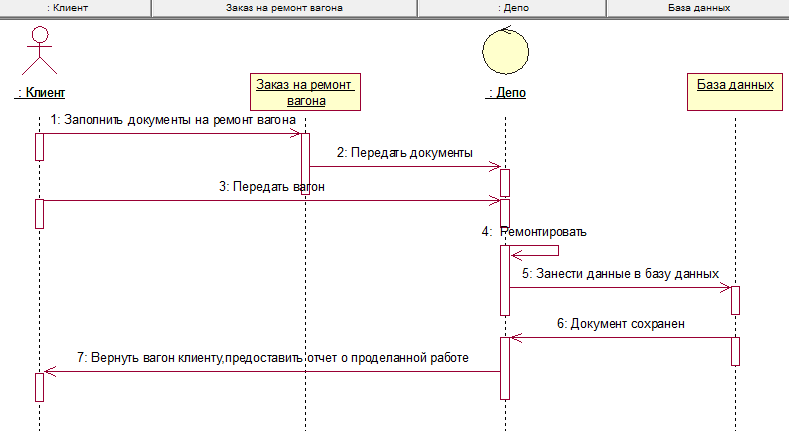
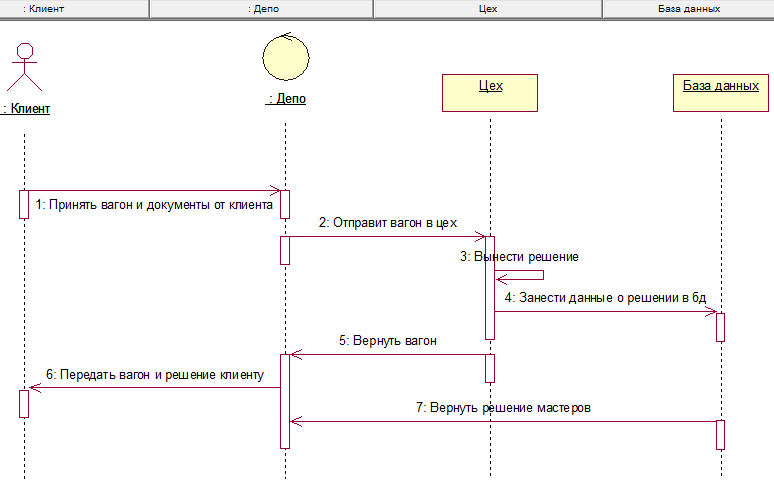


Рис. 2

 Рис.3

Рис.4

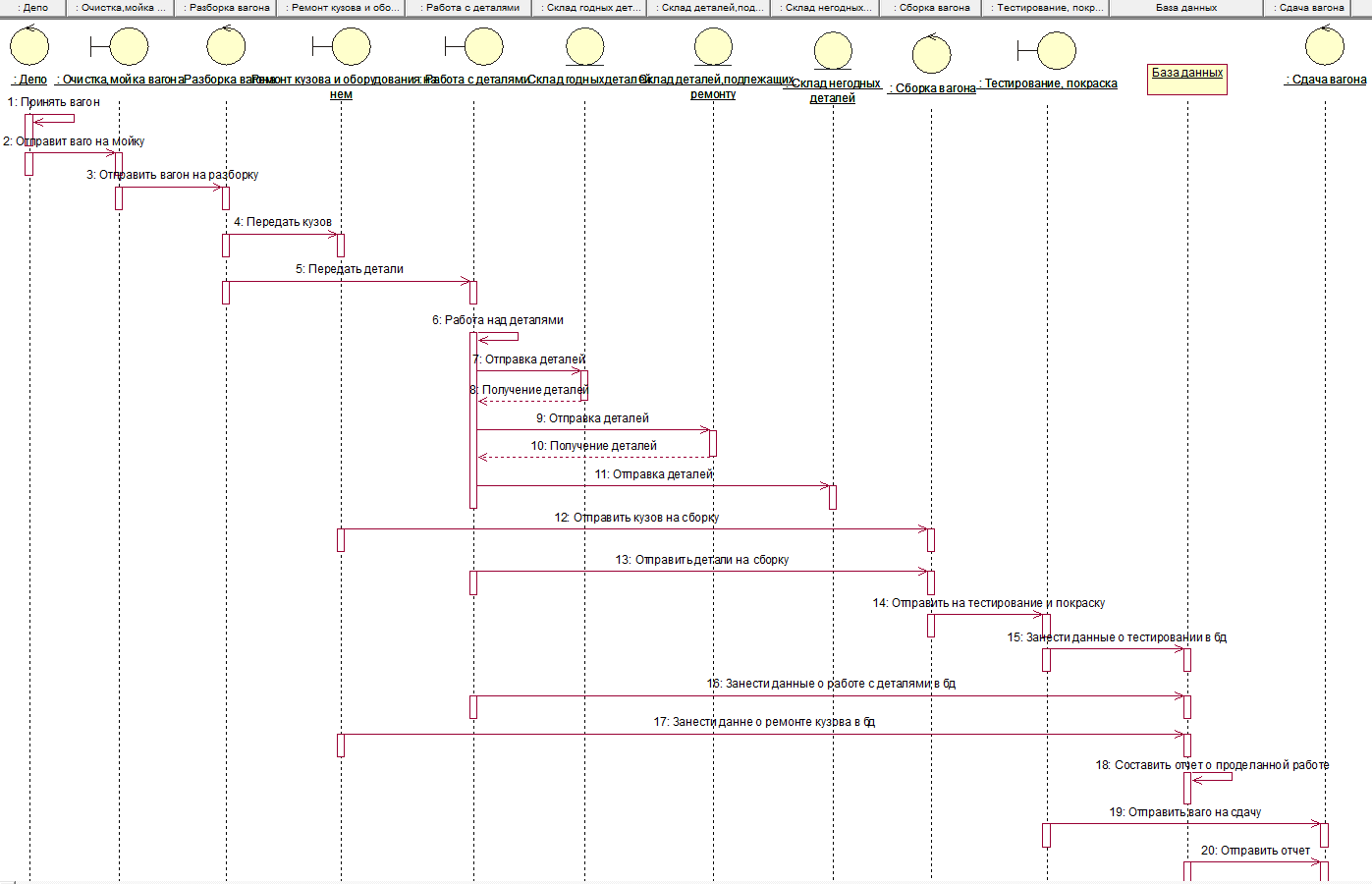


Рис.5

**Диаграмма последовательности** (sequence diagram) — [диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), на которой для некоторого набора объектов на единой временной оси показан жизненный цикл какого-либо определённого объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актёров (действующих лиц) ИС в рамках какого-либо определённого прецедента (отправка запросов и получение ответов).

Альтернативный поток (клиент) рис. 2 и основной поток (клиент) рис.3 практически одинаковы, за исключением нескольких этапов.

1.Клиент заполняет документа на ремонт.

2.Документы передаются в депо

3.Вагон передается в депо

4.Депо выносит вердикт: отказ в ремонте (рис. 2) или же ремонтирует (рис.3)

5.Депо заносит данные в базу данных.

6.База данных возвращает документ в депо

7.Депо возвращает вагон и отчет либо о проделанной работе (рис. 2), либо о причине отказа в ремонте (рис.3).

Альтернативный поток (депо) рис.4

1.Депо получает от клиента вагон и документы

2.Депо отправляет вагон в Цех

3.Цех выносит решение.

4.Решение заносится в базу данных

5.Цех возвращает вагон в Депо

6.База данных возвращает решение

7.Решение и вагон возвращают клиенту

Основной поток (депо) рис.5

1.Депо принимает вагон

2.Вагон отправляют на мойку

3.Затем на разборку в 2 этапа

4.Передаю кузов на ремонт

5.Передают детали

6.Проводят работу над деталями

7,8. Отправка получение деталей со склада «Годных деталей»

9,10. Отправка получение деталей со склада «Детали, подлежащие ремонту»

11. Отправка деталей на склада «Негодных деталей»

12. Кузов отправляют на сборку

13. Детали отправляют на сборку

14. Собранный вагон отправляют на тестирование

15. Заносят результаты тестирования в бд

16.Данные о работе с деталями заносятся в бд

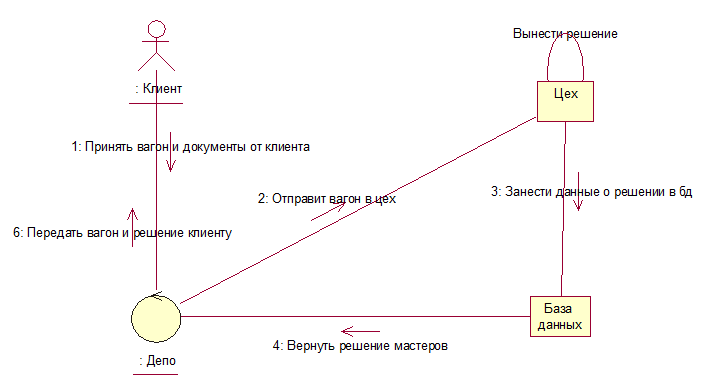
17.Данные о ремонте кузова заносятся в бд

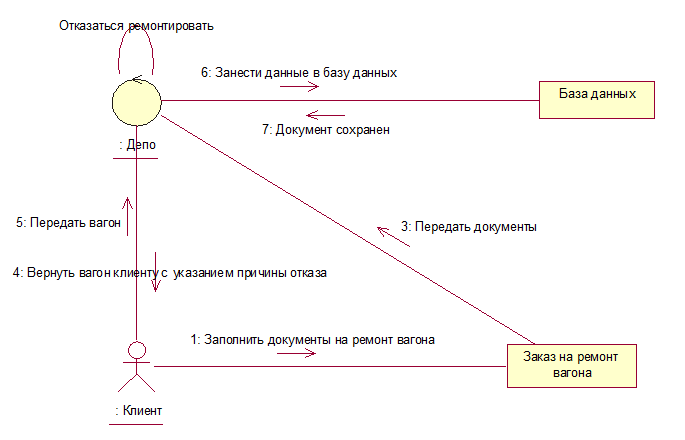
18.Составляется отчет

19.Вагон отправляется на сдачу

20.Отчет отправляется на сдачу

# Диаграмма кооперации

 Рис. 6

 Рис. 7

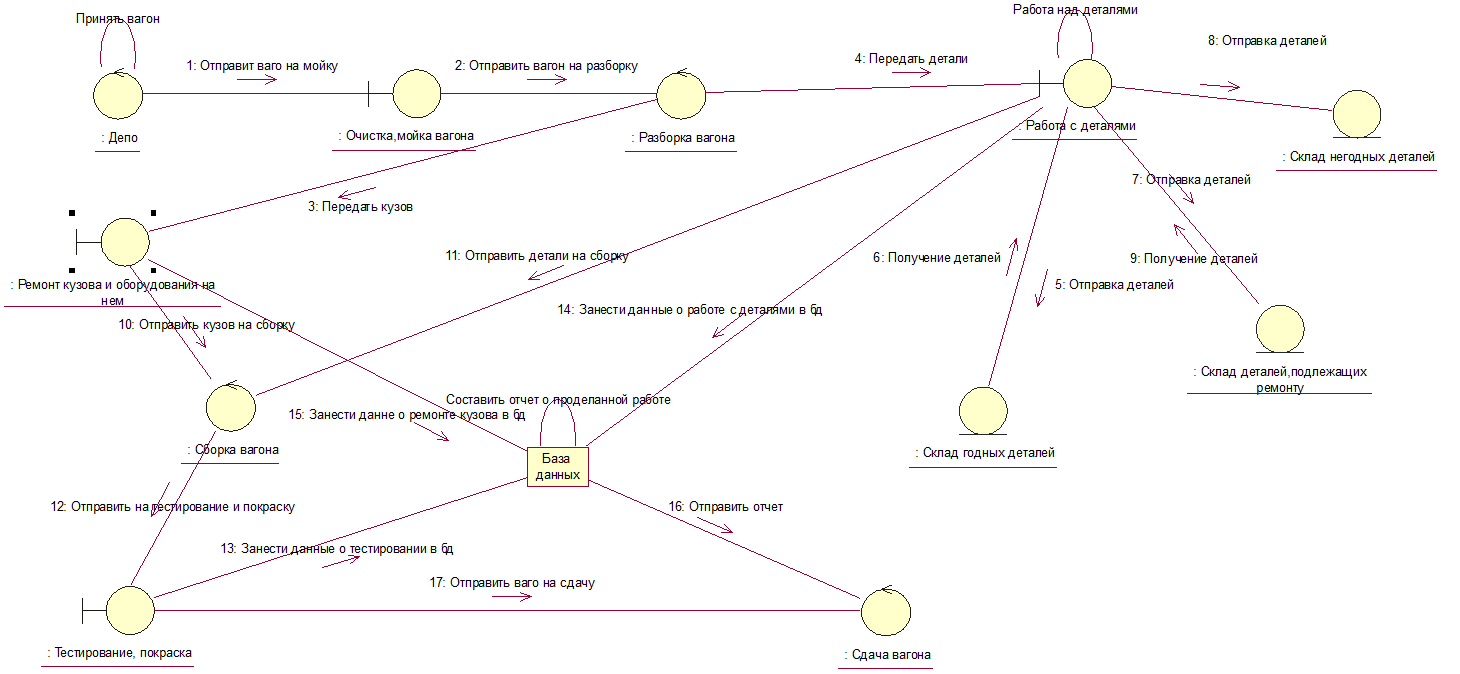
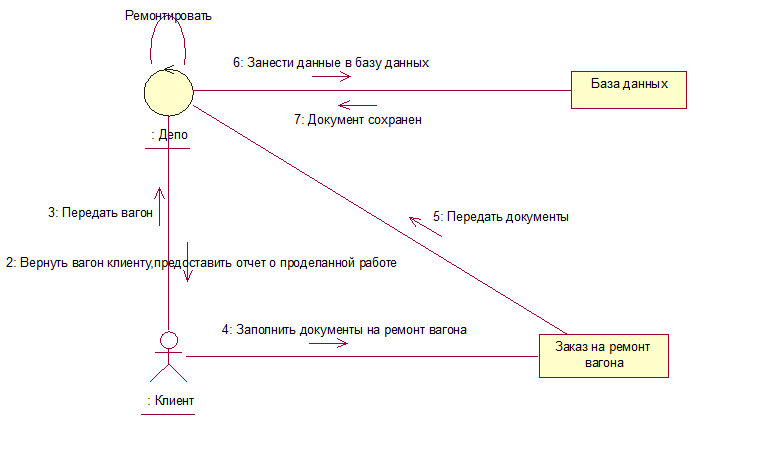


Рис. 8

 Рис. 9

**Диаграмма кооперации** (collaboration diagram) **–**это диаграмма взаимодействий, основное внимание которой уделяется структурной организации объектов, принимающих и отправляющих сообщения. Графически такая диаграмма представляет собой граф из ребер и вершин. Они также отображают поток событий в конкретном сценарии варианта использования. В отличии от диаграмм последовательностей больше внимания заостряют на связях между объектами.

Диаграммы коопераций при основном потоке: рис. 6 и рис. 7

Диаграммы коопераций при альтернативном потоке: рис. 8 и рис. 9

# Диаграмма классов

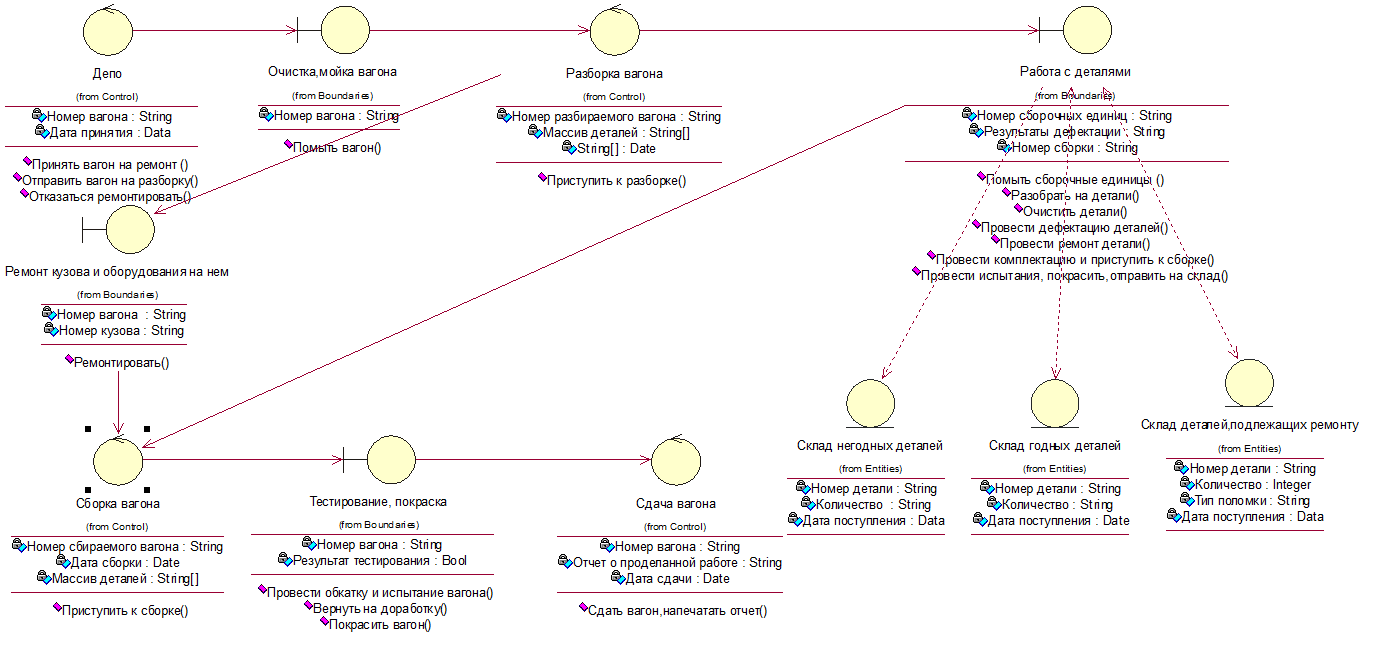


Рис. 10

**Диаграммой классов** в терминологии UML называется диаграмма, на которой показан набор **классов, интерфейсов, коопераций** , а также **связей (отношений)** между ними.

**Атрибутом класса** называется именованное свойство класса, описывающее множество значений, которые могут принимать экземпляры этого свойства.   
  
Класс может иметь любое число атрибутов (в частности, не иметь ни одного атрибута).

**Операцией** класса называется именованная услуга, которую можно запросить у любого объекта этого класса. Операция – это абстракция того, что можно делать с объектом.   
  
Класс может содержать любое число операций (в частности, не содержать ни одной операции). Набор операций класса является общим для всех объектов данного класса.

**Стереотип** – механизм, используемый для создания нового элемента моделирования (для создания новых типов классов). Стереотип класса указывается под его именем и заключается в двойные скобки « stereotype» .

**Класс-сущность** используется для моделирования данных и поведения с длинным жизненным циклом. Этот тип классов может представлять сущности реального мира или внутренние элементы системы.   
  
**Граничные классы** обеспечивают взаимодействие между окружающей средой и внутренними элементами системы.   
  
**Управляющие классы** служат для моделирования последовательного поведения одного или нескольких действующих лиц и координации событий, реализующих заложенное в них поведение.

**Ассоциацией** называется структурная связь, показывающая, что объекты одного класса некоторым образом связаны с объектами другого или того же самого класса. Допускается, чтобы оба конца ассоциации относились к одному классу.

**Кратностью** (multiplicity) роли ассоциации называется характеристика, указывающая, сколько объектов класса с данной ролью может или должно участвовать в каждом экземпляре ассоциации (в UML экземпляр ассоциации называется соединением – link). Наиболее распространенным способом задания кратности роли ассоциации является указание конкретного числа или диапазона. Например, указание «1» говорит о том, что все объекты класса с данной ролью должны участвовать в некотором экземпляре данной ассоциации, причем в каждом экземпляре ассоциации может участвовать ровно один объект класса с данной ролью.

На рис. 10

Сущности: Склады: «Годные детали», «Негодные детали», «Детали, подлежащие ремонту».

Граничные: «Очистка, мойка вагона», «Работа с деталями», «Ремонт кузова и оборудования на нем», «Тестирование, покраска»

Контролы: «Депо», «Разборка вагона», «Сборка вагона», «Сдача вагона»

# Диаграмма состояний

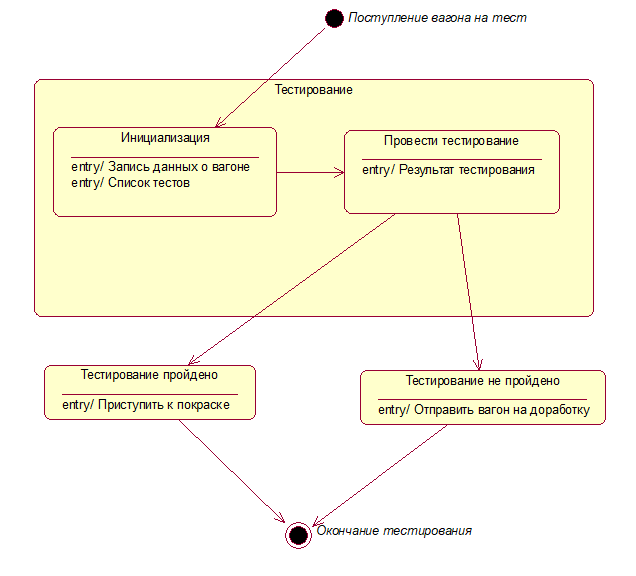


Рис. 11

На диаграмме состояний (рис. 11) отображают жизненный цикл одного объекта, начиная с момента его создания и кончая разрушением.  
  
**Диаграмма состояний** (stateshart diagram) - показывает положение одиночного объекта, события или сообщения, которые вызвают переход из одного состояния в другое, и действия, являющиеся результатом смены состояния.  
  
**Состояние** (state) **–**это некое положение в жизни объекта, при котором он удовлетворяет определенному условию, выполняет некоторое действие или ожидает события.

# Диаграмма компонентов

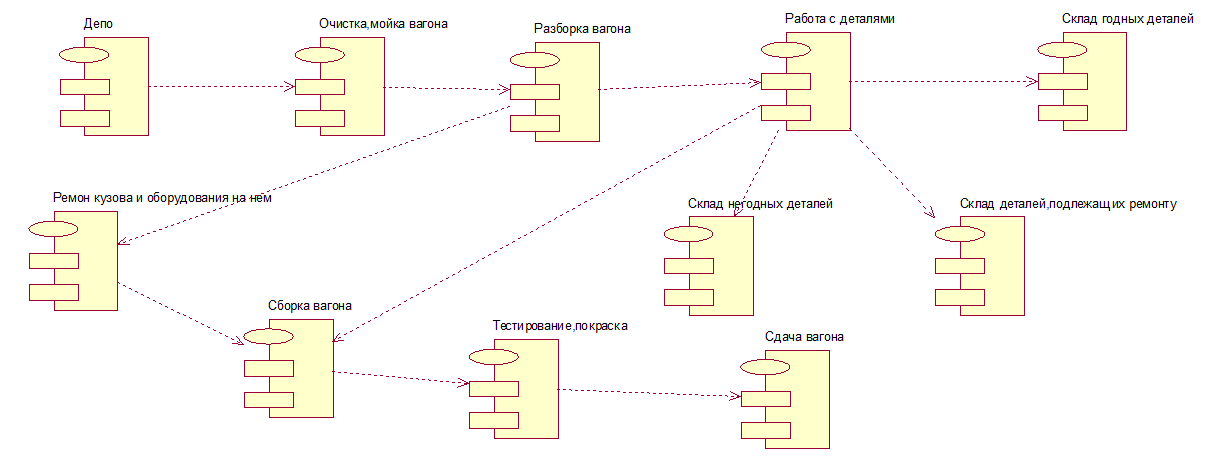


Рис. 12

**Компонентом (**Component) – называется физический модуль кода.

**Диаграммой компонентов** (Component diagram) **–**называется диаграмма (рис. 12), на которой показаны компоненты системы и связи между ними.

# Диаграмма размещения

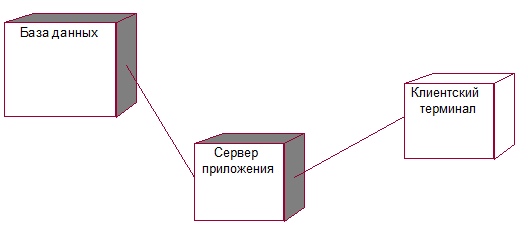


Рис. 13

**Диаграмма размещения** отображает все узлы сети, связи между ними и процессы, выполняющиеся на каждом узле.  
  
**Процессором** (processor) **–**называется любая машина, имеющая вычислительную мощность, т.е. способная производить обработку данных.  
В эту категорию попадают серверы, рабочие станции и другие устройства, содержащие физические процессоры.

**Устройством** (device) **-** называется аппаратура, не обладающая вычислительной мощностью. Например, это устройства ввода-вывода, принтеры, сканеры. Процессоры и устройства называются узлами (nodes) сети.

На рис. 13 показано размещение основных компонентов системы.

В качестве устройства ввода-вывода служит клиентский терминал, куда будет поступать вся основная информация и оттуда она же будет выдаваться.

В качестве процессоров будет сервер приложения (он будет принимать все запросы), база данных (в ней будет хранится весь цикл работ проведенных над вагоном)

# Заключение

Я, Иванов Роман, произвёл объектно-ориентированный анализ выбранного мной варианта задания. Благодаря Rational Rose было произведено проектирование системы мониторинга ремонта ЖД вагонов в депо.

# Использованная литература

1. Описание требований в контексте модели прецедентов

<http://geum.ru/next/art-104426.php>

1. Диаграмма последовательности

<https://ru.wikipedia.org/wiki/Диаграмма_последовательности>