Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Кубанский государственный технологический университет»

Кафедра информационных систем и программирования

**КОЛЛЕКТИВНАЯ РАЗРАБОТКА УПРАВЛЕНИЕ ПРОГРАМНЫМ ПРОДУКТОМ**

**Отчет по лабораторной работе №2**

Выполнил:

Студент 4 курса

группы 19-КБ-ПИ1

Корендюк А.Ю.

Краснодар

2022

**Задание:**

Построить пример модели исходных данных для программного проекта. Использовать унифицированный язык моделирования (UML). Сделать словесное описание модели.

**Выполнение задания:**

Название «Автосервис»

В данной работе была составлена модель исходных данных для программного проекта.

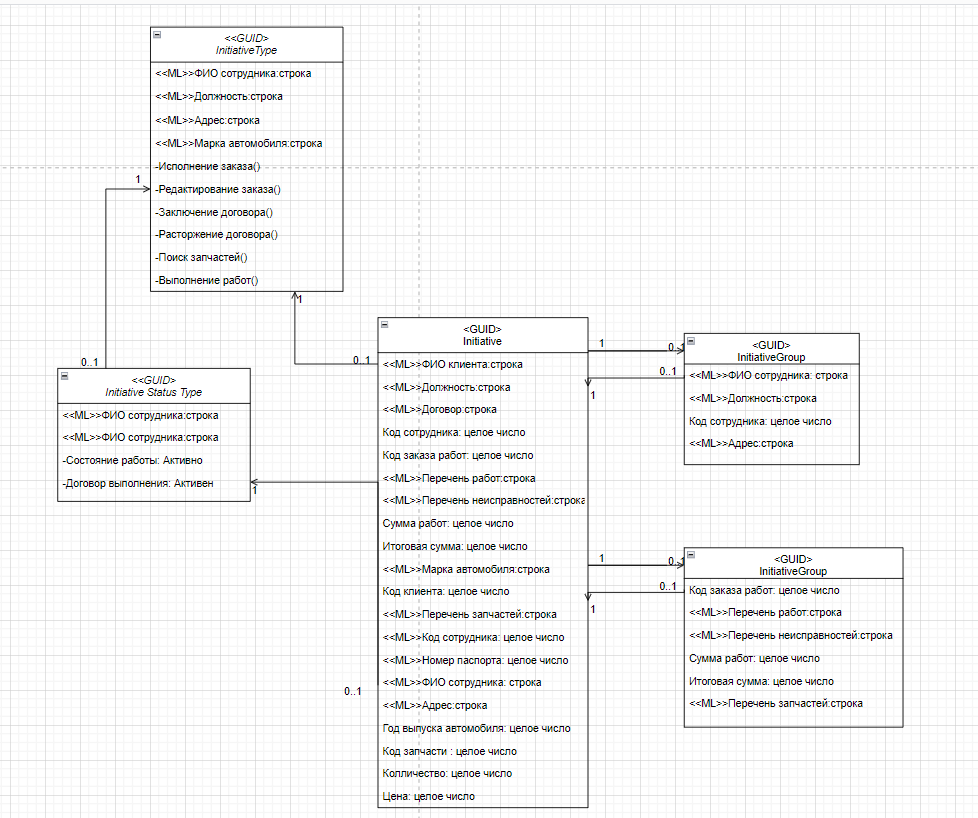


Рисунок 1 – Модель исходных данных Автосервиса

Таблица 1 - Названия элементов и их описания для структуры данных - исходные данные проекта

|  |  |
| --- | --- |
|  |  |
| Название элемента (название поля данных) | Описание |
| Объект (Initiative) | Элемент описывает любую единицу исполнения задания как часть структурного плана проекта. Он также может определять программу и портфель проекта. Результаты, такие как контрольные точки, также могут определяться элементом "Объект", причем используются только данные, значимые для события |
| Группа объектов (InitiativeGroup) | С помощью этого элемента можно соединить объекты в группы. Эти группы, в свою очередь, могут быть подчинены одному объекту. Такая группировка может быть необходимой для создания отчетности или графического изображения планов проекта |
| Тип статуса объекта (InitiativeStatusType) | Элемент описывает различные состояния (статусы), которые могут быть определены для объекта. Таким образом, выражается, в какой фазе жизненного цикла или на каком этапе реализации находится объект. Для каждого типа объекта могут быть определены различные статусы |
| Тип объекта (InitiativeType) | Элемент позволяет определять различные типы объектов. Он позволяет, например, осуществлять дифференцирование на пакеты работ и проекты. Таким образом, пользователь всегда может определить для себя любые типы объектов |

В некоторых из этих классов есть методы которые принадлежат конкретному классу и выполняют определённые функции которые были заданы изначально. Описание элементов представлено в таблице 1.

Представление классов осуществляется с помощи диаграммы классов, которая включает в себя базовые элементы данных классы и атрибуты (рисунок 2).

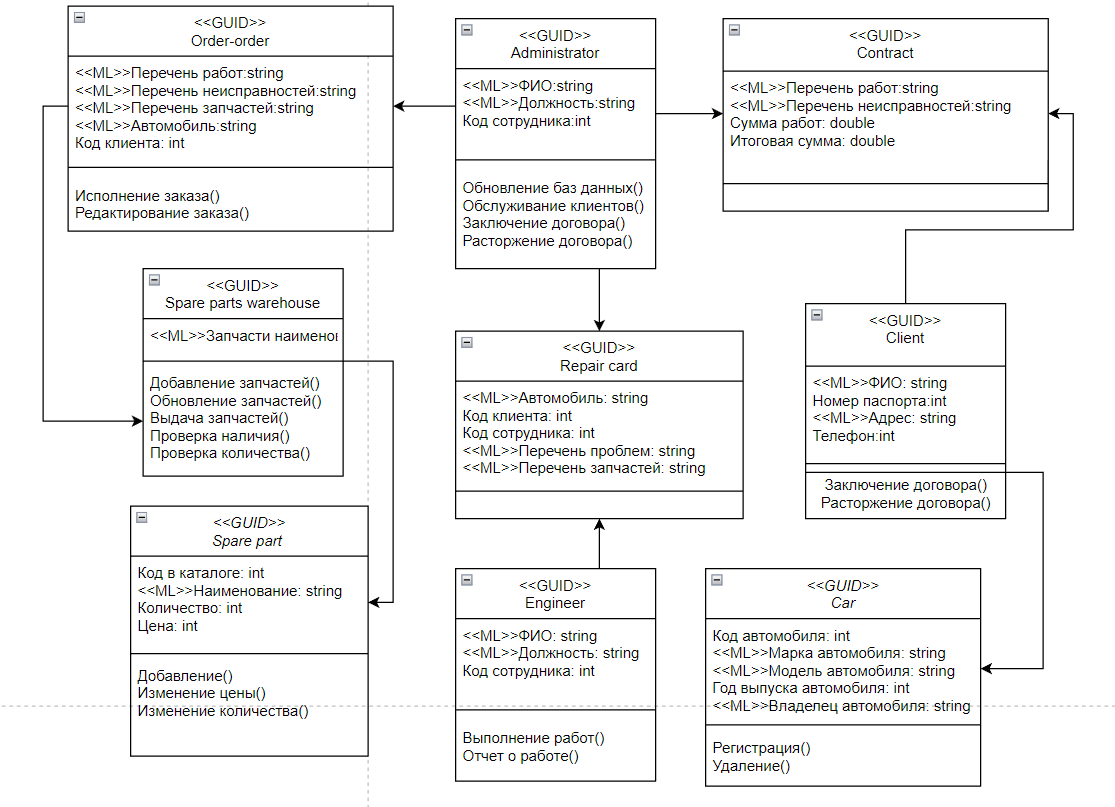


Рисунок 2 – Диаграмма классов предприятия

Таблица 1 – Описание данных

|  |  |
| --- | --- |
| Название элемента (название поля данных) | Описание |
| Spare parts warehouse  (Склад запчастей) | Представляют данные о запчастях которые в данный момент находятся на складе. Главный атрибут: Запчасти наименование |
| Spare part  (Запчасти) | Данные о определенной запчасти и наличие ее на складе. Главный атрибут: Код в каталоге. |
| Repair card  (Карточка ремонта) | Данные заполняемые администратором для ремонта автомобиля, передается инженеру. Главный атрибут: Код клиента |
| Engineer  (Инженер) | Действующее лицо, выполняет работы по карточке ремонта, которую получает от администратора. Главный атрибут: Код сотрудника |
| Car  (Автомобиль) | Содержит метод регистрации и удаления автомобиля. Главный атрибут: код автомобиля |
| Client  (Клиент) | Действующее лицо, заключает и может расторгнуть договор, устно передает администратору пожелания по починке автомобиля. Главный атрибут: Код клиента. |
| Administrator  (Администратор) | Действующее лицо, человек который работает с клиентом через форму договора, выполняет заказ наряд и заполняет карточку ремонта автотранспорта. Главный атрибут: Код сотрудника |
| Order order  (Заказ наряд) | Заказ на поставку запчастей и выдачу их со склада заполняется администратором. Главный атрибут: Код клиента |
| Contract  (Договор) | Данные заполняемые администратором, необходимы для работы с клиентом и составлением его пожеланий по ремонту автомобиля. Главный атрибут: Код клиента |

9 классов данных:э

* Spare parts warehouse(Склад запчастей)
* Spare part(Запчасти)
* Repair card(Карточка ремонта)
* Engineer(Инженер)
* Car(Автомобиль)
* Client(Клиент)
* Administrator(Администратор)
* Order order(Заказ наряд)
* Contract(Договор)

В некоторых из этих классов есть методы которые принадлежат конкретному классу и выполняют определённые функции которые были заданы изначально. Описание классов представлено в таблице 1.

Представление модели данных осуществляется с помощи диаграммы классов, которая включает в себя базовые элементы данных классы и атрибуты (рисунок 2).