Группа ПИЖ-о-б-20-1
Симоненко Александр Сергеевич

Выполнил UML диаграмма по ПЭТ проекту:

UML диаграммы для продукта с реализацией упрощенной работы ремонтного центра.

Цель программы – автоматизация выполнения различных видов действий в центре по ремонту различной бытовой техники для повышения эффективности работы менеджеров, удобства мастеров и сбора статистических значений для возможного перераспределения ресурсов центра. Данная программа позволяет вести учет сотрудников, удобную работу с клиентами и их техникой, управлять каталогом услуг, работать со складом деталей, а также быстро формировать и изменять заявки на ремонт.

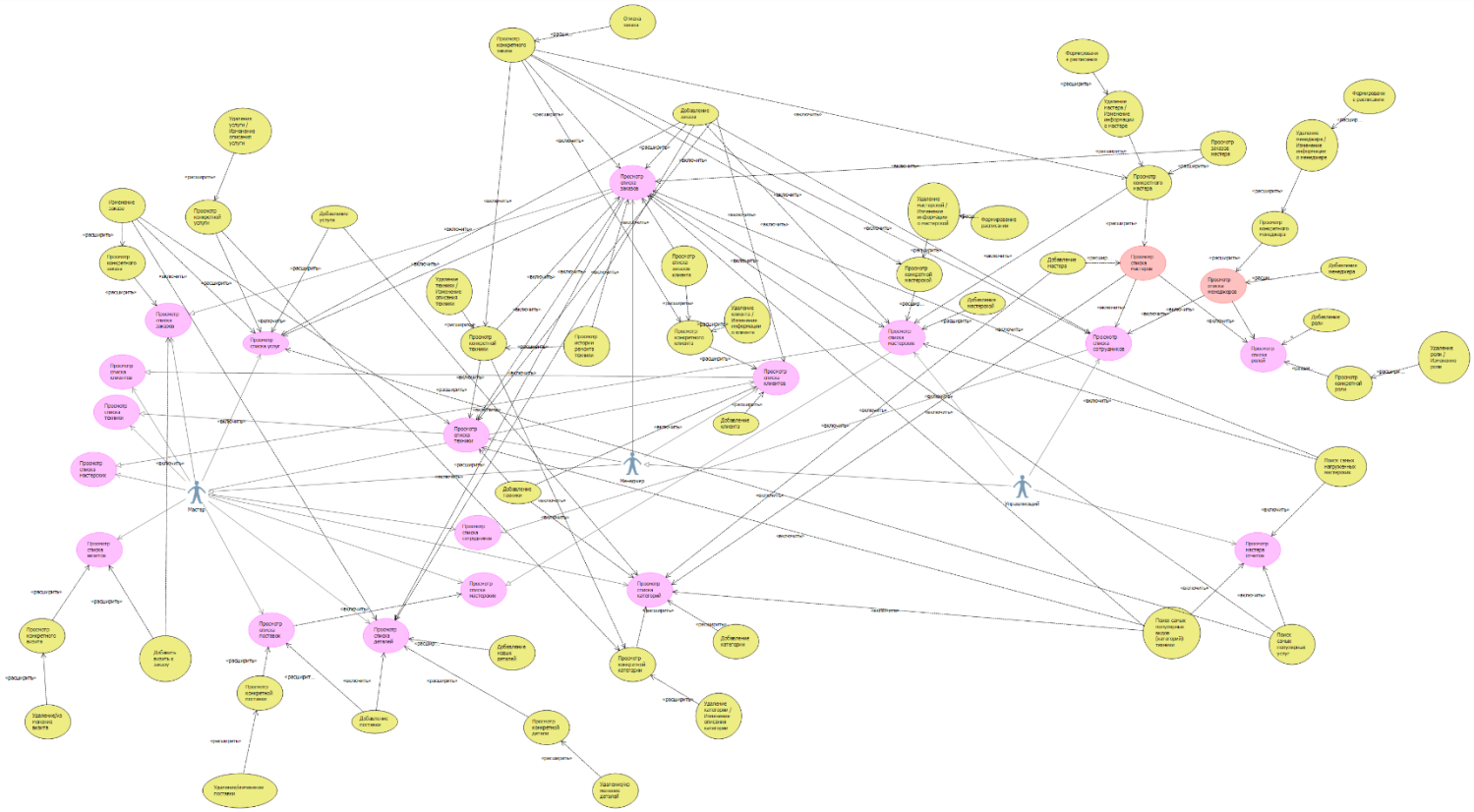

Рисунок 1.1 – Диаграмма прецедентов

Группа ПИЖ-о-б-20-1
Симоненко Александр Сергеевич

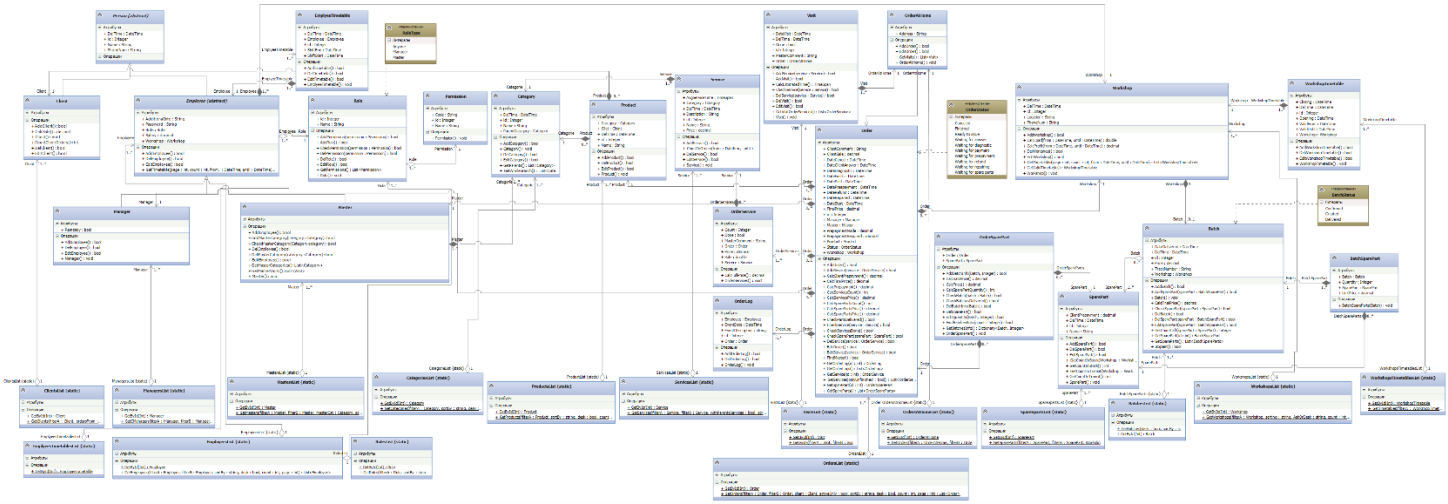

Рисунок 1.2 – Диаграмма классов

Список классов

- Person - абстрактный класс, представляющий собой человека. От него наследуются Client и Employee
- Client - класс, представляющий собой клиента сервисного центра
- Employee - Абстрактный класс, представляющий собой работника сервисного центра. От него наследуются Manager и Master. У всего сотрудников есть расписание, которое можно получить с помощью метода GetTimetable()
- Manager - класс, представляющий собой менеджера сервисного центра
- Master - класс, представляющий собой мастера сервисного центра. Мастеры разделяются по категориям техники, которую они могут умеют обслуживать. Для получения этих категорий предназначен метод GetMasterCategories()
- EmployeeTimetable - класс, представляющий собой расписание работника
- Role - класс, представляющий собой роль сотрудника. У каждой роли есть ряд разрешений Permission, которые вызываются с помощью метода GetPermissions(). Сотруднику с данной ролью доступны действия, перечисленные в этой самой коллекции.

- Permission - класс, представляющий собой определенное действие в система. С помощью этого класса вместе с Role регулируются права пользователей в система.
- Category - класс, представляющий собой определенную категорию техники. Благодаря нему можно группировать технику Product клиентов и контролировать, может ли мастер Master обслужить ту или иную технику. В качестве одного из атрибутов содержит родительскую категорию, для выстраивания иерархии
- Product - класс, представляющий собой технику клиента Client
- Service - класс, представляющий собой оказываемую услугу. У услуги есть категория Category, к которой она относится. Время выполнения предназначено для примерного планирования времени выезда Visit, чтоб минимизировать вероятность конфликта выездов и опозданий.
- Order - класс, представляющий собой заказ на ремонт. Содержит коллекцию деталей OrderSpareParts и услуг OrderServices, которые можно получить методами GetSpareParts() и GetServices()
- OrderServices - класс, связывающий услугу и заказ. Нужен для того, чтоб была возможность регулировать количество одной и той же услуги в заказе (например, замена двух конфорок на плите), тонкой настройки скидки на конкретную услугу в заказе (например, акция на 100% скидку на диагностику), а также хранить цену за услугу в момент ее добавления к заказу
- OrderAtHome - класс, представляющий собой заказ на ремонт на дому. Наследуется от Order. Содержит коллекции визитов, которые можно получить методом GetVisits()
- Visit - класс, представляющий собой выезд на дом при заказе на дому
- SparePart - класс, представляющий собой описание запчасти. Атрибут ClientPrepayment определяет, нужна ли предоплата в заказ, в которых требуется эта деталь

- OrderSpareParts - класс, представляющий собой запчасть в заказе. Содержит словарь, в котором указанно, из какой поставки была взята запчасть и в каком количестве. Благодаря чему есть возможность выбрать, из каких конкретных поставок будет использована запчасть и в каком количестве. Если же такая тонкая настройка не требуется, то детали можно подобрать автоматически, вызвав метод FindBatchesAuto(), указав, какое количество запчасть данного типа надо. Весь набор поставок можно получить с помощью метода GetBatchesInfo(). Само описание детали хранится в атрибуте SparePart
- Batch - класс, представляющий собой поставку с деталями BatchSpareParts. Коллекцию деталей можно получить с помощью метода GetSpareParts()
- BatchSpareParts - класс, связывающий детали и поставку. Благодаря нему мы можем указать, какое количество деталей выбранного типа SparePart нужно заказ и цену на них (цена не хранить в самом класс SparePart, т.к. в разных поставках цена может отличаться)
- Workshop - класс, представляющий собой мастерскую (филиал сервисного центра). Содержит в себе расписания работы филиала WorkshopTimetable, получаемые с помощью метода GetTimetables(). Также содержит методы для расчета статистики филиала
- WorkshopTimetable - класс, представляющий собой расписание работы филиала. Т.к. расписание у филиалов меняется редко, оно хранится не на каждый день, а на промежутки времени. Атрибут ValidFrom показывает, с какого числа действует расписание, а ValidUntil до какого числа.
- ...List (static) - данные статические классы (ClientsList, OrdersList, ServicesList и т.д.) отвечают за сортировку экземпляров классов, а также в некоторых из них реализован расчет статических данных.

Группа ПИЖ-о-б-20-1
Симоненко Александр Сергеевич

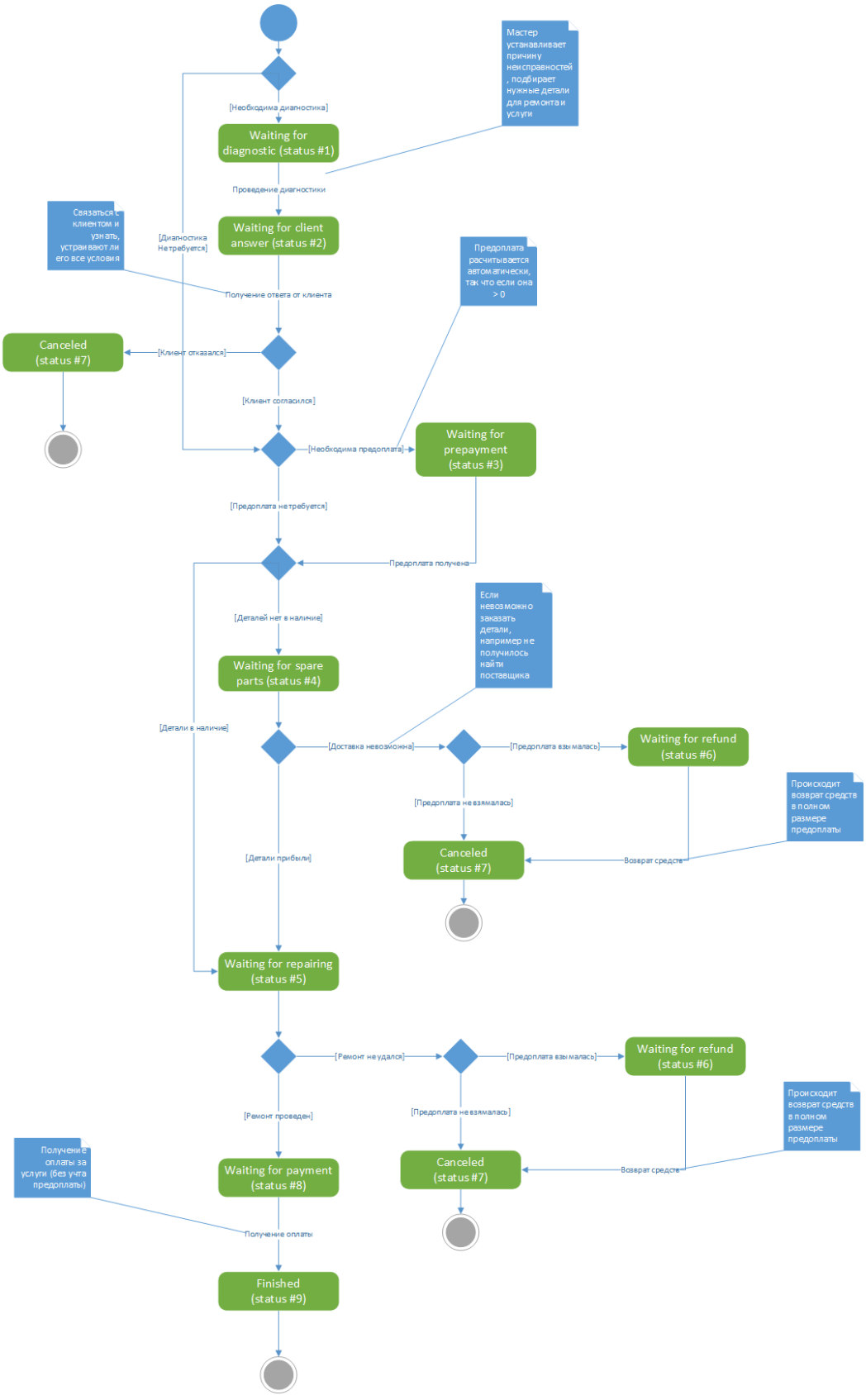

Рисунок 1.3 – Диаграмма состояний