UML диаграммы

1. Обобщение (наследование) — класс Employee расширяет класс Man

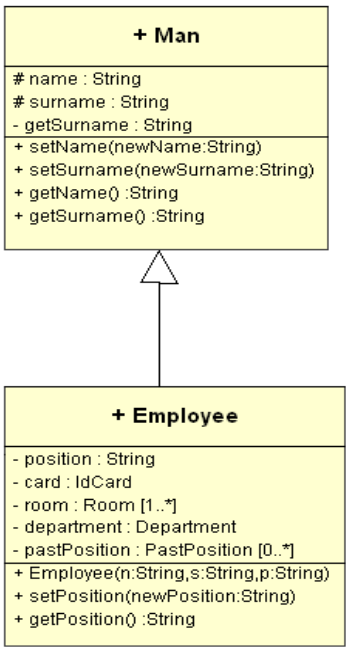

Рис. 2 — Отношение обобщения

2. Ассоциация — отношение между объектами-экземплярами класса

2.1 Бинарная ассоциация — связь один к одному, одному объекту класса 1 может соответствовать только один объект класса 2

В модель добавили класс «IdCard», представляющий идентификационную карточку(пропуск) сотрудника. Каждому сотруднику может соответствовать только одна идентификационная карточка, мощность связи 1 к 1.

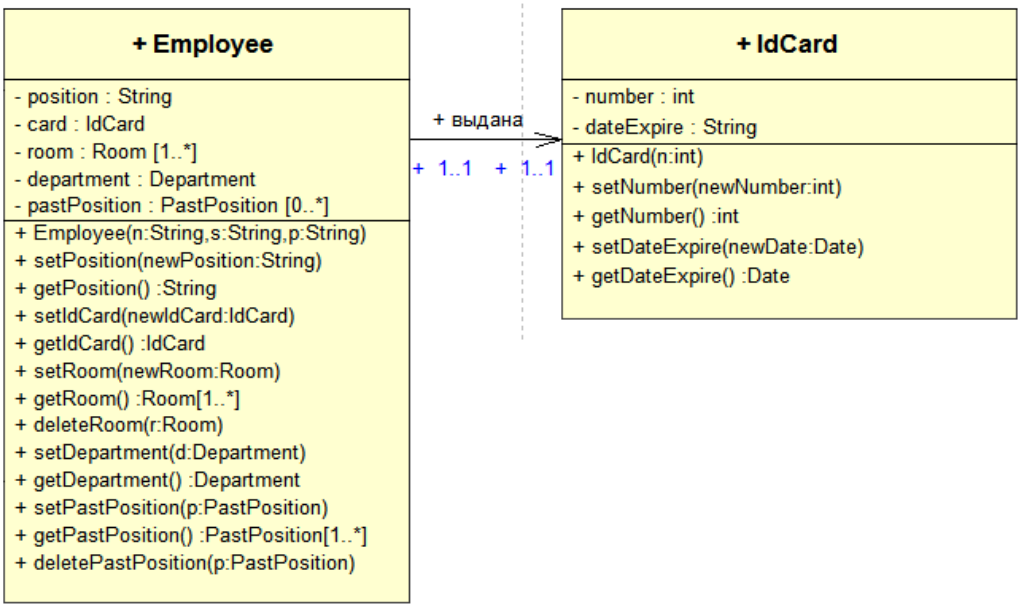

Рис. 3 — Бинарная ассоциация

2.2 N-арная ассоциация — связь один ко многим

Представим, что в организации положено закреплять за работниками помещения. Добавляем новый класс Room.
Каждому объекты работник(Employee) может соответствовать несколько рабочих помещений. Мощность связи один-ко-многим.

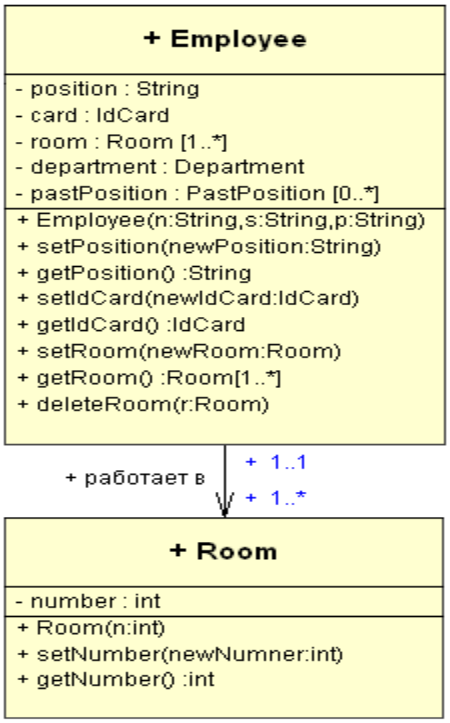

Рис. 4 — N-арная ассоциация

2.3 Агрегация — объединение

Введем в модель класс Department(отдел) — наше предприятие структурировано по отделам. В каждом отделе может работать один или более человек. Можно сказать, что отдел включает в себя одного или более сотрудников и таким образом их агрегирует. На предприятии могут быть сотрудники, которые не принадлежат ни одному отделу, например, директор предприятия.

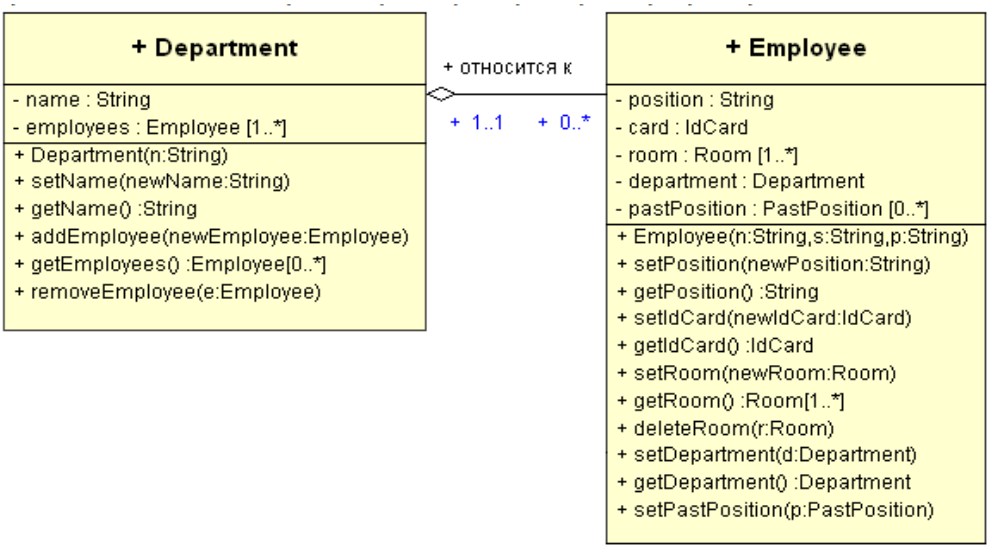

Рис. 5 — Агрегация

2.3.1 Композиция

Предположим, что одним из требований к нашей системе является требование о том, чтоб хранить данные о прежней занимаемой должности на предприятии.
Введем новый класс «pastPosition». В него, помимо свойства «имя»(name), введем и свойство «department», которое свяжет его с классом «Department».

Данные о прошлых занимаемых должностях являются частью данных о сотруднике, таким образом между ними связь целое-часть и в то же время, данные о прошлых должностях не могут существовать без объекта типа «Employee». Уничтожение объекта «Employee» должно привести к уничтожению объектов «pastPosition».

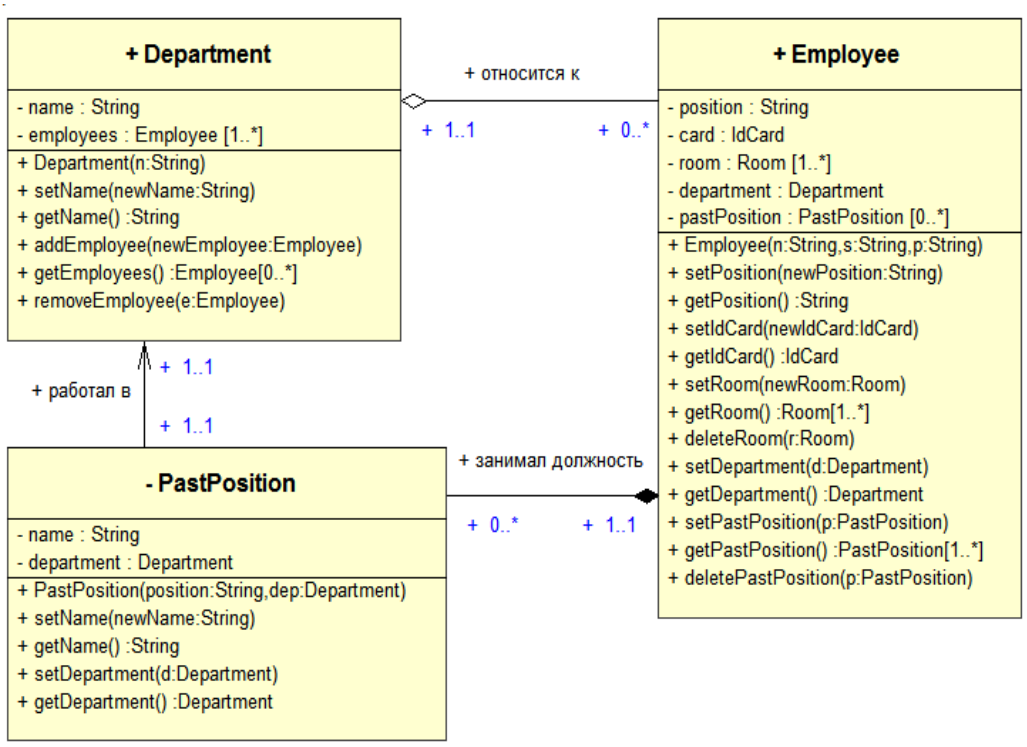

Рис. 6 — Композиция

3. Зависимость

Для организации диалога с пользователем введем в систему класс «Menu». Встроим один метод «showEmployees», который показывает список сотрудников и их должности. Параметром для метода является массив объектов «Employee». Таким образом, изменения внесенные в класс «Employee» могут потребовать и изменения класса «Menu».

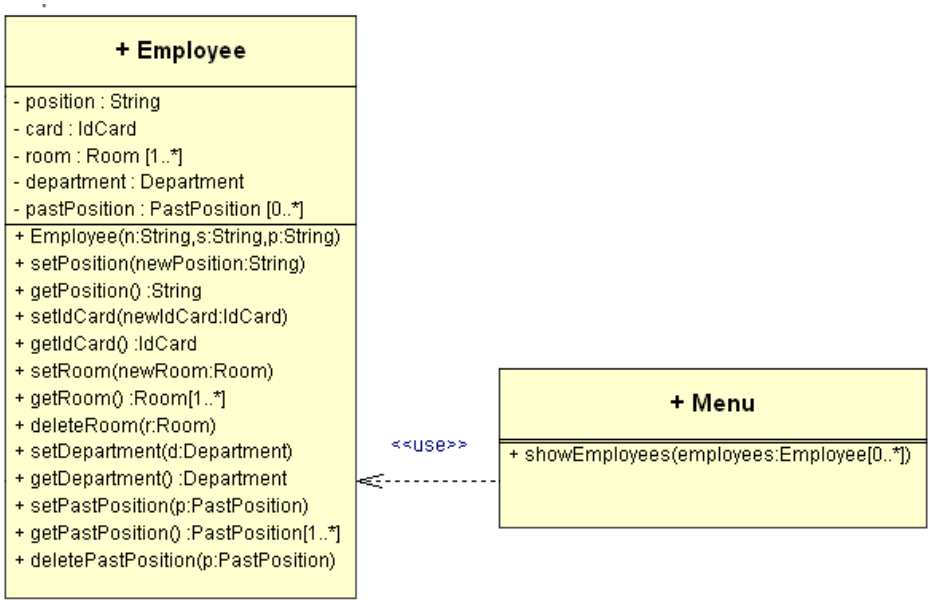

Рис. 7 — Зависимость

4. Реализация

Реализация, как и наследование имеет явное выражение: объявление интерфейса и возможность его реализации каким-либо классом.

Для демонстрации отношения «реализация» создадим интерфейс «Unit». Если представить, что организация может делиться не только на отделы, а например, на цеха, филиалы и т.д. Интерфейс «Unit» представляет собой самую абстрактную единицу деления. В каждой единице деления работает какое-то количество сотрудников, поэтому метод для получения количества работающих людей будет актуален для каждого класса реализующего интерфейс «Unit».

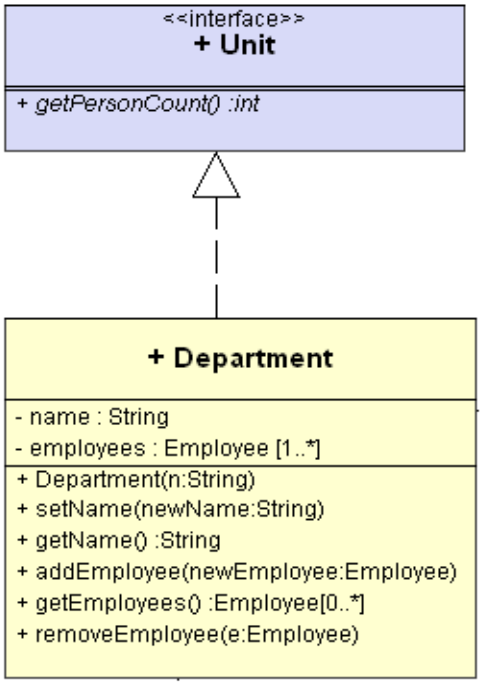

Рис. 8 — Реализация