

## 3.3 Логическая модель проектируемой подсистемы

### 3.3.1 Диаграмма классов

Изучить структуру иерархии классов системы и построить диаграмму классов анализа рассматриваемой системы. Построить диаграмму классов рассматриваемой системы.

#### Пример выполнения работы

Правила:

- При выделении классов анализа следует учитывать тот факт, что они являются обобщенными (укрупненными) сущностями, которые в дальнейшем подлежат уточнению и возможному разбиению на несколько более мелких классов.

- Для выделения классов сущностей необходимо определить все реальные либо воображаемые объекты, имеющие существенное значение для рассматриваемой предметной области, информация о которых подлежит хранению. При этом из спецификаций вариантов использования следует выделить все объекты, которые могут существовать независимо от других. Например, объект «билет» является независимой сущностью, потому что любой билет существует независимо от того, знаем мы его номер, стоимость или нет. Т. е. при выделении классов-сущностей действуют те же правила, что при построении концептуальной модели БД).

- Для каждого актера следует предусмотреть, как минимум, один граничный класс в целях организации интерфейса между ним и системой. Аналогично для каждого класса сущности, как правило, должен быть граничный класс – ведь по каждому объекту класса сущности должна быть предусмотрена возможность просмотра, ввода и/или корректировки информации через определенную форму ввода/вывода или чтения/записи через определенный интерфейс.

- Для управления, обеспечения взаимодействия и координации работы объектов, реализующих одну из функций системы (обычно, вариант использования), необходимо предусмотреть, как минимум, один управляющий класс. Как правило, взаимодействие между граничным классом и классом сущности происходит через управляющий класс.

- В целях облегчения восприятия специфики связей между классами рекомендуется использовать отношения агрегации, композиции и обобщения.

- При разработке диаграммы основное внимание должно быть уделено определению и детализации классов сущностей, управляющих и граничных классов, обеспечивающих взаимодействие с внешними системами. Граничные классы, обеспечивающие взаимодействие с пользователями, не

требуют излишней детализации до уровня отдельного поля ввода или ниспадающего списка, так как современные среды программирования обладают богатыми возможностями по быстрому созданию пользовательского интерфейса.

Правила составления диаграммы классов:

- 3-5 методов в классе – оптимальное количество;
- классы не должны быть изолированными;
- не создавать много мелких классов (1-2 метода);
- не объединять все в крупные классы (> 6 методов);
- избегать всемогущих классов («...System» «...Controller»);
- избегать глубокой иерархии наследования.

Пример диаграммы классов анализа представлен на рисунке 1.

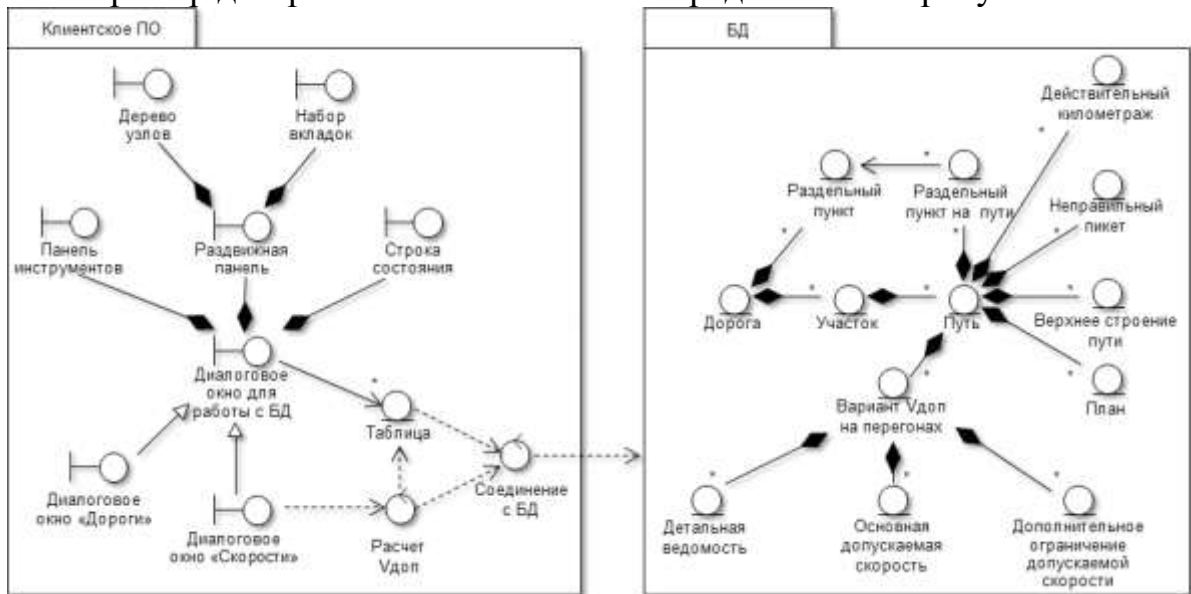


Рис. 1. Фрагмент диаграммы классов анализа

Рассмотрим пример системы обработки заказов.

Ключевые абстракции -- основные понятия предметной области. В системе обработки заказов можно выделить следующие ключевые абстракции: Order (данные о заказе), OrderItem (данные о позиции заказа), ArticleOfFurniture (данные о предмете мебели), Client (данные о заказчике), User (учётная запись пользователя системы), LoginForm (форма входа в систему), MainMenu (форма главного меню), OrderForm (форма заказа), ASAccess (форма-посредник, реализующий протокол взаимодействия с Бухгалтерской системой), OrderController, LoginController и DataAccess (экземпляры управляющего класса).

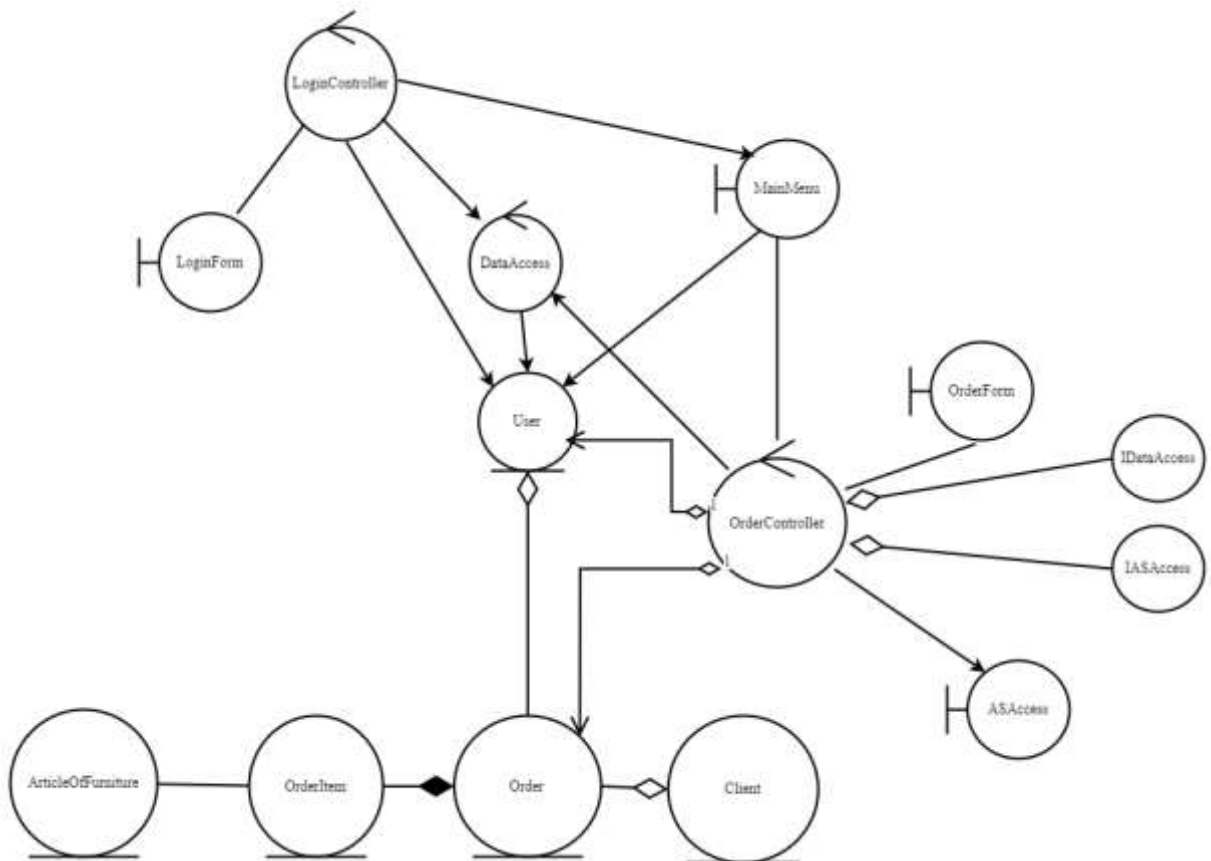


Рисунок 2 – Диаграмма классов анализа системы обработки заказов

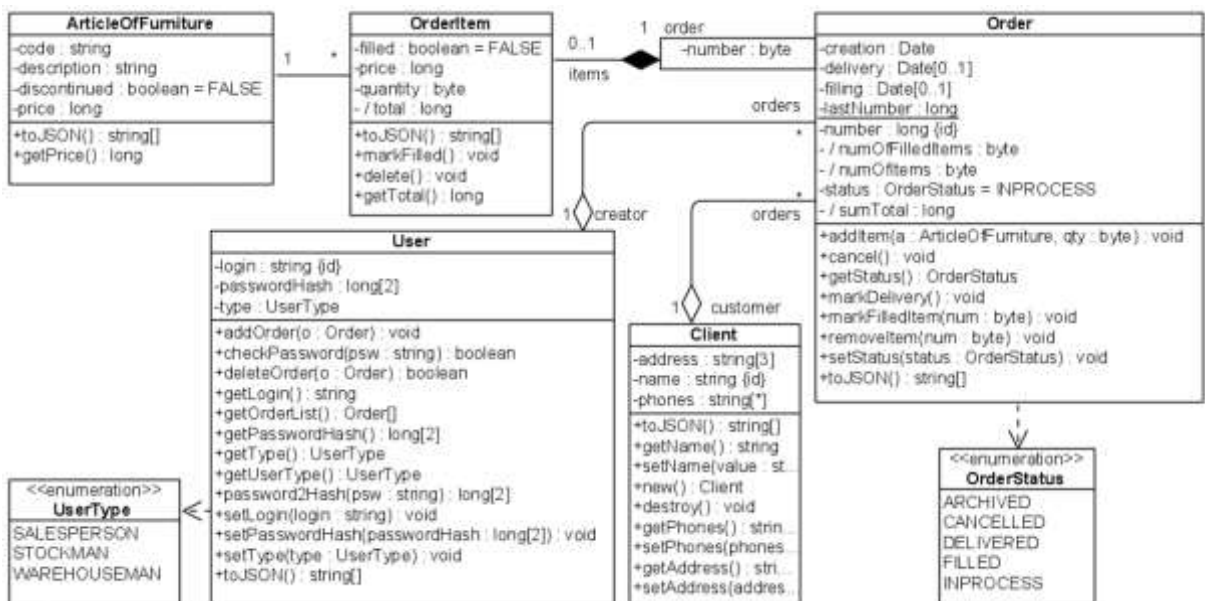


Рисунок 3 – Фрагмент диаграммы классов

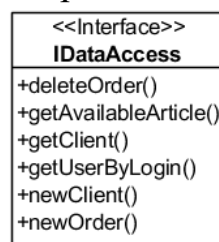
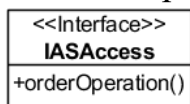


Рисунок 4 - UML-диаграмма классов Interfaces

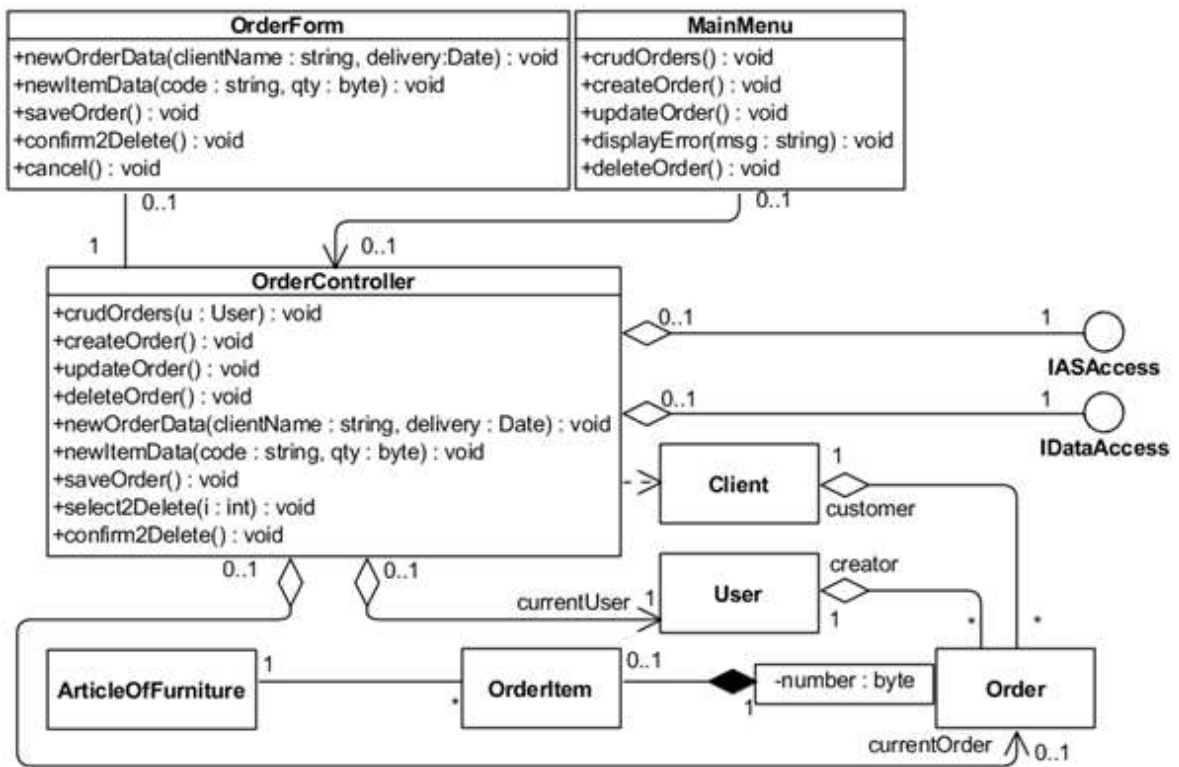


Рисунок 5 – Фрагмент диаграммы классов