# Лабораторная работа 2. Отношения классов. Ассоциации. Наследование.

## Задание 1. Ассоциация.

**Композиция**. Предполагает создание вложенных объектов в конструкторе класса-контейнера. (Не стоит создавать продавцов вместе с магазином.) Если логически должна быть связь 1-∞, для композиции создать минимум 2 объекта (можно использовать массив).

Для главного класса-контейнера разработать несколько перегруженных конструкторов:

1) без параметров,

2) с параметрами для полей этого класса

3) с параметрами для полей этого класса и подчиненного класса-композиции

**Агрегация или ассоциация (в зависимости от логики предметной области)**. Предполагает связь по ссылке. Ссылка может быть null.

Здесь возможны варианты:

1. Создать массив объектов в классе-контейнере (массив машин в гараже). Изначально все ссылки null.
2. Вместо того, чтобы делать массив ссылок в главном классе, можно в каждом подчиненном объекте сделать ссылку на главный класс (например, каждая машина имеет ссылку на свой гараж – эта ссылка может быть null).
3. То и другое – при добавлении машины к гаражу, в гараже меняется соответствующее место в массиве, у машины меняется ссылка на гараж. Соответствующие методы должны быть в том и другом классе и каждый метод должен изменять объекты обоих классов.

Спроектировать UML-диаграммы классов согласно варианту индивидуального задания. В каждом классе предусмотреть поля (2-4), конструкторы, свойства (2-4) и методы (1-2). Между парами классов спроектировать отношения ­– композицию (1) и агрегацию (1). Продемонстрировать работу классов.

**Варианты заданий.**

1. Факультет – Кафедра – Студент.
2. Автомобиль – Двигатель – Гараж.
3. Аэропорт – Взлетная полоса – Самолет.
4. Танкер – Танк – Порт.
5. Гостиница – Номер – Клиент.
6. Город – Достопримечательность – Турист.
7. Магазин – Секция – Продавец.
8. Поезд – Вагон – Груз.
9. Интернет-оператор – Услуга – Абонент.
10. Отдел – Сотрудник – Организация.

## Задание 2. Наследование.

Выделить в предметной области 2-3 варианта сущности (например, "Геометрическая фигура", "Точка", "Треугольник"), отличающиеся несколькими полями и методами. Каждый класс имеет поля, свойства и методы.

Спроектировать UML-диаграммы классов.

**Базовый** класс для вашей иерархии объявите **абстрактным**. Он должен содержать **абстрактные методы** и **методы с реализацией**.

Один из наследников должен **перегружать** метод родителя.

Один из классов должен содержать **виртуальный метод**, который **переопределяется** в одном наследнике и **не переопределяется** в другом.

Продемонстрировать работу всех объявленных методов.

Продемонстрировать **вызов конструктора родительского класса** при наследовании.

Продемонстрировать вызов метода родительского класса при его **скрытии**.

Создать класс, **закрытый для наследования**.

Примеры методов: вывод полей класса, изменение числовых полей класса (скорости, дохода, численности штата и т.д.), изменение текстовых полей класса (переименование), какая-либо динамика (полет, ремонт, переезд, открытие/закрытие и.д.)

**Номер варианта выбирается по желанию!**

1. Геометрические фигуры
2. Здания
3. Транспорт
4. Автомобили
5. Товары
6. Игровые юниты
7. Компьютерная техника
8. Программное обеспечение
9. Игрушки
10. Налоги
11. Банковские счета
12. Врачи
13. Лекарства
14. Мебель
15. Фильмы
16. Одежда
17. Техника
18. Еда
19. Алкогольные напитки
20. Коктейли
21. Фрукты или овощи
22. Десерты
23. Космические объекты
24. Спорт
25. Животные
26. Растения
27. Научные дисциплины