

### Практическая работа №3. Наследование и LINQ-запросы.

Реализовать иерархию классов для выбранного базового типа. В качестве базового класса можно использовать вариант из таблицы, либо предложить свой собственный базовый класс и иерархию наследников (**но отсутствующие в таблице**).

Вариант для таблицы вычисляем по формуле:  $V = (\text{int}(c1) + \text{int}(c2)) \% 9$ , где  $c1$  и  $c2$  – **вторая** буква в фамилии и имени на английском языке в верхнем регистре. В программе выводить номер варианта.

Вариант	Базовый класс
0	Person
1	Animal
2	Vehicle
3	Instrument
4	Figure
5	Game
6	FlyingObject
7	SwimmingObject
8	Document

Помимо базового класса реализовать не менее **5 производных классов** и, по крайней мере, 3 уровня в иерархии (н-р, класс С наследует В, а класс В наследует А).

В каждом классе ввести **закрытые поля и открытые свойства** доступа к полям.

В иерархии классов задействовать ключевые слова: **virtual, override, abstract, static, sealed**.

Реализовать возможность работы со списком объектов через консольное меню со следующей функциональностью:

- добавление новых объектов разных типов иерархии;
- удаление объектов;
- сортировка объектов по выбранному полю;
- подсчёт количества объектов каждого типа;
- поиск объекта по какому-либо полю;
- поиск всех объектов, удовлетворяющих условию (н-р, «найти все инструменты с количеством струн не меньше 4-х»);
- поиск максимума или минимума: объект, который обладает наибольшей или наименьшей характеристикой (н-р, «найти млекопитающее животное с наибольшим весом»);
- вычисление агрегированной характеристики по объектам определенного типа (н-р, «суммарный вес по всем самолётам», «средний возраст преподавателей»).

Для реализации сортировки использовать метод Sort и интерфейсы. Для реализации поиска объектов и вычисления агрегированных характеристик использовать LINQ-операторы.