

### Лабораторная работа №3 Наследование. Простое и множественное наследование

В каждом варианте использовать наследование от еще одного/нескольких классов, которые предложены в условии или нет. Не запрещается использовать дополнительные классы, которые не указаны в условии. В каждом создаваемом классе должно быть поле с ID экземпляра. Предусмотреть автоматическое и корректное заполнение полей класса данными. Реализация и объявление методов классов в разных файлах. Перегрузка операторов не обязательна, но желательна по усмотрению. Для хранения данных использовать только один список/массив.

**ДОПЫ СДАЮТСЯ ВМЕСТЕ С ОСНОВНЫМ ЗАДАНИЕМ**

1) Создать классы для работы **сайта**. Классы для реализации: Пользователь, модератор, редактор, администратор. Также создать список с информацией сайта, например, пусть хранится информация о продаваемых товарах/услугах. У каждого пользователя независимо от роли по началу должна быть возможность войти на сайт для последующей работы.

2) Создать классы для **игры**. Классы для реализации: Entity, NPC, Enemy, Player, Administrator. Предусмотреть возможность взаимодействия с другими классами. Например, при взаимодействии игрока с NPC можно будет с ним поговорить, при взаимодействии администратор с остальными, можно изменить свойства экземпляра (здоровье, урон, опыт, диалоги и тд). У каждого пользователя независимо от роли по началу должна быть возможность войти в игру, используя логин и пароль.

3) Создать классы для работы **предприятия**. Классы для реализации: Сотрудник, Менеджер, Тестировщик, Разработчик, Администратор, Директор. Добавить также список с информацией о производимых/произведенных товарах. Внутри должна храниться информация о разработчике и менеджере. Добавить возможность изменять статус товара (в разработке, производится, есть на складе, нет в наличии) администраторам. Также добавить возможность ставить задачи на выполнение (по истечению времени задача считается выполненной и соответственно, изменяется статус).

4) Создать классы для работы с **календарем**. Классы для реализации: Встреча, Задача, Напоминание. Базовый для них класс должен хранить данные о дате, дне недели. Для каждого класса предусмотреть уникальные функции взаимодействия с выполнением. Например, Встреча будет считаться выполненной по истечению времени выделенному на встречу и только потом можно будет отметить, посещена встреча или нет.

5) Создать классы **предметов** для игры. Классы для реализации: Item, Weapon, Resource, Potion. Класс Item содержит название предмета, максимальное количество в слоте инвентаря и возможный вес и функции для работы с самим слотом непосредственно. Каждый остальных класс содержит уникальные методы для использования предмета в зависимости от класса.

---

#### ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ:

Создать функции для кодирования и декодирования строки используя "Rail Fence Cipher".  
"WEAREDISCOVEREDFLEEATONCE" -> "WECRLTEERDSOEFEAOCAIVDEN"

W	E	C	R	L	T	E
E	R	D	S	O	E	E
A	I	V	D	E	N	