**Курс:** Шаблоны проектирования приложений

**Тема:** Модуль 04 Принципы проектирования

*Баллы:*

Задача: **Разработка системы управления пользователями**

Вам нужно разработать простую систему управления пользователями для небольшой компании. Система должна позволять добавлять, удалять и обновлять информацию о пользователях. Каждый пользователь должен иметь имя, email и роль (например, "Admin" или "User").

**Условия**

1. **Минимальные требования**:
   * Создайте класс **User**, который содержит свойства для хранения имени, email и роли.
   * Создайте класс **UserManager**, который будет управлять коллекцией пользователей (например, используя List<User>). В UserManager реализуйте методы для добавления (AddUser), удаления (RemoveUser) и обновления (UpdateUser) пользователей.
2. **Применение принципа YAGNI**:
   * Избегайте добавления функциональности, которая не требуется для текущей задачи. Например, не добавляйте методы для поиска пользователей по нескольким критериям, если такой функционал не нужен сейчас.
   * Реализуйте только те методы, которые действительно необходимы для работы с пользователями.
   * Не создавайте абстракции, такие как интерфейсы для пользователей или менеджеров, если сейчас они не требуются.
3. **Применение принципа KISS**:
   * Поддерживайте код как можно более простым. Избегайте сложных конструкций, ненужных абстракций и избыточного функционала.
   * Реализуйте только те свойства и методы, которые решают текущие задачи, не усложняя код.
   * Не вводите дополнительные проверки и условия, если они не нужны. Стремитесь к линейной структуре кода и минимальному числу вложенных условий и циклов.
4. **Применение принципа DRY**:
   * Избегайте дублирования кода. Если вы обнаружите повторяющийся код в нескольких местах, вынесите его в отдельный метод или класс.
   * Стремитесь к тому, чтобы каждый фрагмент кода существовал только в одном месте, и все операции по его изменению выполнялись централизованно.
   * Подумайте, как можно объединить схожие операции, такие как добавление и обновление пользователя, чтобы минимизировать дублирование.

**Требования:**

* YAGNI: Реализуйте задачу, придерживаясь принципа YAGNI. Не добавляйте функциональность, которая не требуется для выполнения минимальных требований.
* KISS: Поддерживайте код простым и понятным, избегайте излишних абстракций, сложных конструкций и ненужных проверок.
* DRY: Минимизируйте дублирование кода. Если какая-то логика повторяется, вынесите ее в отдельный метод или класс.
* Код должен быть легко читаемым и понятным.
* Если потребуется добавить новую функциональность в будущем, сделайте это с минимальными изменениями в существующем коде.

Эта задача помогает вам практиковать принципы YAGNI, KISS и DRY, избегая избыточной функциональности, сохраняя код простым, и минимизируя дублирование.