**Чек-лист для проверки ДЗ**

**Упрощаем код: YAGNI, DRY, KISS в Java-проектах**

*(заполняется спикером урока)*

**1. Проверка нарушения принципа DRY:**

- Проверьте, указал ли студент места, где есть повторение кода, особенно на уровне строк подключения к базе данных.

**Пример из кода (нарушение DRY):**

try (Connection connection = DriverManager.getConnection( ... );

PreparedStatement preparedStatement = connection.prepareStatement(query)) {

...

}

**Примерно ожидаемый комментарий студента:**

"Этот блок кода повторяется. Рекомендую создать утилитный метод для получения соединения с БД."

- Проверьте, есть ли замечания по повторному использованию блоков кода, таких как подключение к БД и выполнение запросов SQL.

**Пример из кода (нарушение DRY):**

public Article ca(Article article) {

// Подключение к БД повторяется в каждом методе

...

}

**Примерно ожидаемый комментарий студента:**

"Код подключения к БД повторяется в каждом методе работы с данными. Рекомендуется вынести в отдельный приватный метод или класс."

- Проверьте, отмечены ли места, где повторяется логика хеширования пароля.

**Примерно ожидаемый комментарий студента:**

"В каждом месте, где требуется хеширование пароля, вызывается hashPassword. Рекомендуется вкладывать хэш только при создании или изменении пароля, а не при каждом запросе данных пользователя."

**2. Проверка нарушения принципа YAGNI:**

- Указал ли студент на неиспользуемые методы или поля классов?

**Пример из кода (нарушение YAGNI):**

public static boolean isAdministrator = false;

**Примерно ожидаемый комментарий студента:**

"Поле isAdministrator объявлено, но нигде не используется. Рекомендуется удалить эту переменную, если она не нужна для текущего функционала."

**3. Проверка нарушения принципа KISS:**

- Проверьте, нашел ли студент части кода, которые переусложнены, и предложены ли более простые решения.

**Пример из кода (нарушение KISS):**

public List<User> gau() {

List<User> users = new ArrayList<>();

// Простой запрос превращен в переусложненный списочный цикл

...

return users;

}

**Примерно ожидаемый комментарий студента:**

"Этот метод можно упростить, используя Java 8 Streams для преобразования ResultSet в список пользователей."

Ниже приведены конкретные примеры из предоставленного кода, которые студент должен был бы заметить и прокомментировать:

- Многократное использование строки подключения к базе данных в каждом методе нарушает принцип DRY. Студент должен был предложить использование конфигурационного файла или глобальных переменных.

- Каждый метод работы с базой данных содержит повторяющийся код подключения к базе данных, что нарушает DRY.

- Методы ca, gai, ga, ua, da, cu, gubi, gubu, gau, uu, du, cur, cp в классе D могут иметь более понятные имена, что сделает код проще для понимания (нарушение KISS).

Убедитесь, что студент предложил конкретные улучшения для каждой проблемы: использование паттернов проектирования, таких как DAO или Repository для доступа к данным, общие методы для работы с базой данных и т.д.

**Само ДЗ для вашего понимания:**

Пришло время закрепить полученные теоретические знания практикой. Оцените приложенный проект, идентифицируя элементы, которые нарушают основы DRY, KISS и YAGNI. Отметьте ваши замечания непосредственно в коде, комментируя их в соответствующем pull request'е. Это улучшит ваши навыки критического анализа и поможет углубить понимание этих важных принципов.

Проект: <https://github.com/EkaterinaMikrukova/OOP-Java>

[Подробная инструкция к ДЗ - Упрощаем код: YAGNI, DRY, KISS в Java-проектах](https://docs.google.com/document/d/1uYH2k7oSzMzNUm5mPvuYPiUq-oXa6Gmerjes-1dDLQE/edit)