FB.UC\_USELESS\_CONDITION

Это условие всегда дает тот же результат, что и значение задействованной переменной, которое было сужено ранее. Вероятно, имелось в виду что-то другое или условие можно убрать.

Таблица №1 – Основные характеристики детектора FB.UC\_USELESS\_CONDITION

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Code Quality Issue |
| CWE | --- |
| CVE | --- |
| Возможные последствия | * Избыточный код * Усложнение понимания логики * Ложное чувство защиты |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  // Некорректно: бесполезное условие  boolean flag = true;    if (flag == true) { // Избыточная проверка  System.out.println("Flag is true");  }    // Другой пример  String text = "constant";  if (text != null && "constant".equals(text)) { // Первая проверка избыточна  System.out.println("Text matches");  }  } | public static void correctTest() {  // Корректно: упрощенные условия  boolean flag = true;    if (flag) { // Простая проверка  System.out.println("Flag is true");  }    // Более эффективная проверка строки  String text = "constant";  if ("constant".equals(text)) { // Безопасно и эффективно  System.out.println("Text matches");  }    // Пример с необходимыми проверками  String input = null;  if (input != null && input.length() > 0) { // Здесь проверка оправдана  System.out.println("Valid input");  }  } |