FB.UC\_USELESS\_CONDITION\_TYPE

Это условие всегда дает один и тот же результат из-за диапазона типов задействованной переменной. Вероятно, имелось в виду что-то другое или условие можно убрать.

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

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Code Quality Issue |
| CWE | --- |
| CVE | --- |
| Возможные последствия | * Избыточные проверки типов * Усложнение читаемости кода * Ложное чувство безопасности |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  // Некорректно: бесполезная проверка типа  String text = "example";    if (text instanceof String) { // Всегда true  System.out.println("Redundant check: " + text.length());  }    // Другой пример  Integer number = 10;  if (number != null && number instanceof Integer) { // Вторая проверка избыточна  System.out.println("Double check: " + number);  }  } | public static void correctTest() {  // Корректно: только необходимые проверки  String text = "example";  System.out.println("Direct use: " + text.length());    // Проверка нужна только когда тип неочевиден  Object obj = "test";  if (obj instanceof String) { // Оправданная проверка  String str = (String) obj;  System.out.println("Valid length: " + str.length());  }    // Для nullable объектов  Integer number = null;  if (number != null) { // Достаточно одной проверки  System.out.println("Number: " + number);  }  } |