FB.UM\_UNNECESSARY\_MATH

Этот метод использует статический метод из java.lang.Math для постоянного значения. Результат этого метода в этом случае может быть определен статически, и быстрее, а иногда и точнее, если использовать константу.

Таблица №1 – Основные характеристики детектора FB.UM\_UNNECESSARY\_MATH

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Performance Issue |
| CWE | --- |
| CVE | --- |
| Возможные последствия | Избыточные вычисления при использовании математических операций; снижение производительности в критических участках кода; увеличение времени выполнения операций |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  // Некорректно: избыточные математические операции  double value = Math.abs(-5.0) \* Math.signum(-5.0); // Всегда даст -5.0  System.out.println("Inefficient value: " + value);    int x = 10;  int y = Math.max(x, x); // Бессмысленная операция  System.out.println("Redundant max: " + y);  } | public static void correctTest() {  // Корректно: упрощенные вычисления  double value = -5.0; // Прямое присваивание  System.out.println("Optimized value: " + value);    int x = 10;  int y = x; // Прямое использование значения  System.out.println("Simplified value: " + y);  } |