FB.URF\_UNREAD\_FIELD

Ошибка "unread field" указывает на то, что в вашем коде есть объявление поля (переменной класса), но это поле не используется (т.е., оно "не читается"). Это может быть признаком лишних или неиспользуемых данных в вашем коде, что может привести к избыточности и понижению читаемости.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Наличие неиспользуемых полей в классе увеличивает сложность кода без добавления полезной функциональности. |
| Комментарий | Рекомендуется удалять неиспользуемые поля, чтобы улучшить читаемость и поддержку кода. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  class MyClass {  private String unusedField = "Неиспользуемое поле";  public void doSomething() {  System.out.println("...");  }  }  MyClass myClass = new MyClass();  myClass.doSomething();  } | public static void correctTest() {  class MyClass {  public void doSomething() {  System.out.println("...");  }  }  MyClass myClass = new MyClass();  myClass.doSomething();  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Удалите объявление поля**, если оно действительно нигде не используется в классе. Это самый простой и эффективный способ устранить проблему.
2. **Проверьте, не было ли поле предназначено для использования в будущем.** Если поле планируется использовать позже, оставьте его, но добавьте комментарий с объяснением его назначения.
3. **Убедитесь, что поле не используется случайно в другом месте класса с опечаткой в имени.** Внимательно просмотрите весь код класса.
4. **Рассмотрите возможность сделать поле локальной переменной метода**, если его использование ограничено одним методом и не требует хранения состояния объекта.
5. **Если поле является константой, рассмотрите возможность объявления его как static final**, если оно может быть полезно на уровне класса, даже если текущий экземпляр его не использует.
6. **Если поле используется только для отладки, удалите его после завершения отладки.** Или оберните его использование условной компиляцией (например, с флагом отладки).
7. **В случае сериализации, проверьте, действительно ли поле не нужно сохранять.** Если поле не влияет на состояние объекта после десериализации, его можно пометить как transient.