Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение
высшего образования «Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ
Центр ускоренного обучения

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

по дисциплине «Конструирование программного обеспечения»

Тема: Статический анализ кода

Студенты гр. РИВ-400027у — Л.А.Кайгородова, О.В.Дрон, Д.И.Кудинов, В.И.Пинтак, Я.В.Козлов

Преподаватель — С.И.Тимошенко, доц., к.т.н

Екатеринбург 2023

Оглавление

1 Постановка задачи 3
2 Анализ поставленной задачи 4
3 Анализ полученных результатов 7

1 Постановка задачи

Проверить с помощью выбранного статического анализатора кода наличие проблем у библиотеки colt (из папки «Материал для лабораторной N 4». Результаты работы с пояснениями проблем включить в отчет. При обнаружении более десяти проблем, описать только наиболее критичные.

2 Анализ поставленной задачи

Для выполнения данной лабораторной работы требуется подключение библиотеки colt.

После подключения был запущен проект colt и проверен с помощью SpotBugs, результаты проделанной работы представлен на рисунке 1.

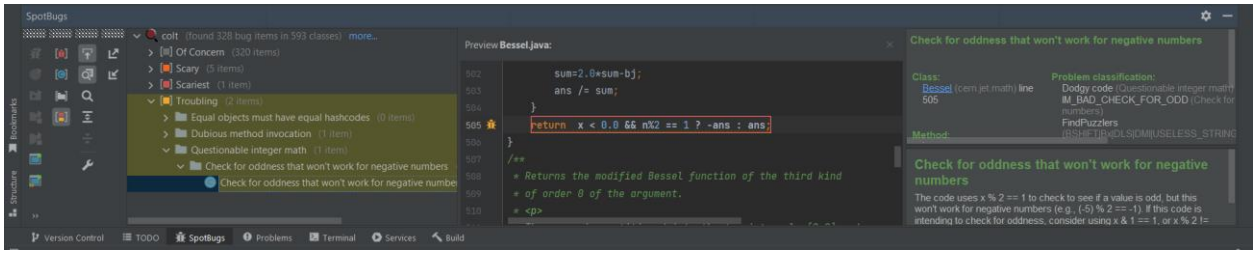

Рисунок 1 – работа SpotBugs с проектом colt

StopBugs проверил данный проект, не ряд ошибок:

- Check for oddness that won't work for negative numbers;

Проверка на нечетность, которая не будет работать для отрицательных чисел. Код использует x % 2 == 1, чтобы проверить, является ли значение нечетным, но это не будет работать для отрицательных чисел (например, (-5) % 2 == -1). Если этот код предназначен для проверки на нечетность, рассмотрите возможность использования x & 1 == 1 или x % 2 != 0.

- Random object created and used only once;

Случайный объект, созданный и использованный только один раз. Этот код создает объект java.util.Random, использует его для генерации одного случайного числа, а затем отбрасывает объект Random. Это создает случайные числа посредственного качества и неэффективно. Если возможно, перепишите код так, чтобы объект Random создавался один раз и сохранялся, а каждый раз, когда требуется новое случайное число, вызывайте метод существующего объекта Random для его получения.

Если важно, чтобы сгенерированные случайные числа нельзя было угадать, вы не должны создавать новый случайный номер для каждого случайного числа; значения слишком легко угадываются. Вместо этого вам следует настоятельно рассмотреть возможность использования

java.security.SecureRandom (и избегать выделения нового SecureRandom для каждого необходимого случайного числа).

- new IllegalArgumentException() not thrown;

Исключение создано и удалено, а не выброшено. Этот код создает объект исключения (или ошибки), но ничего с ним не делает.

- Field SimpleLongArrayList.size masks field in superclass cern.colt.list.AbstractLongList;

Класс определяет поле, которое маскирует поле суперкласса. Этот класс определяет поле с тем же именем, что и видимое поле экземпляра в суперклассе. Это сбивает с толку и может указывать на ошибку, если методы обновляют или обращаются к одному из полей, когда им нужно другое.

- Bad comparison of nonnegative value with 0;

Плохое сравнение неотрицательного значения с отрицательной константой или нулем. Этот код сравнивает значение, которое гарантированно неотрицательно, с отрицательной константой или нулем.

- Repeated conditional test;

Повторные условные тесты. Код содержит условную проверку, выполняемую дважды, одну сразу после другой (например, x == 0 || x == 0). Возможно, второе вхождение должно быть чем-то другим (например, x == 0 || y == 0).

- Useless increment in return from apply(int);

Бесполезное увеличение в операторе возврата. Этот оператор имеет возврат, такой как return x++; / вернуть x--;. Увеличение/уменьшение постфикса не влияет на значение выражения, поэтому это увеличение/уменьшение не имеет никакого эффекта. Убедитесь, что это утверждение работает правильно.

- Found reliance on default encoding: new java.io.FileReader(String).

Использование кодировки по умолчанию. Обнаружен вызов метода, который выполняет преобразование байта в строку (или строки в байт) и

предполагает, что кодировка платформы по умолчанию подходит. Это приведет к тому, что поведение приложения будет отличаться на разных платформах. Используйте альтернативный API и явно укажите имя набора символов или объект набора символов.

3 Анализ полученных результатов

В данной лабораторной работе была осуществлена проверка проекта colt на наличие проблем у проекта colt. SpotBugs нашел ряд ошибок.

Данная работа позволила изучить работу с анализаторами кода и применить знания на практических примерах, также позволила изучить работу с ошибками, на которые указывает анализатор когда.