**­­­­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**

# 1 Постановка задачи

Проверить с помощью выбранного статического анализатора кода наличие проблем у проекта library (из папки «Материал для лабораторной N 4»). Проект требует подключения библиотеки jsr305-2.0.0.jar из той же папки. Результаты работы с пояснениями проблем включить в отчет.

**2 Анализ поставленной задачи**

Для выполнения данной лабораторной работы требуется подключение библиотеки jsr305-2.0.0.jar., для подключения было достаточно положить его в папку lib директории Java.

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

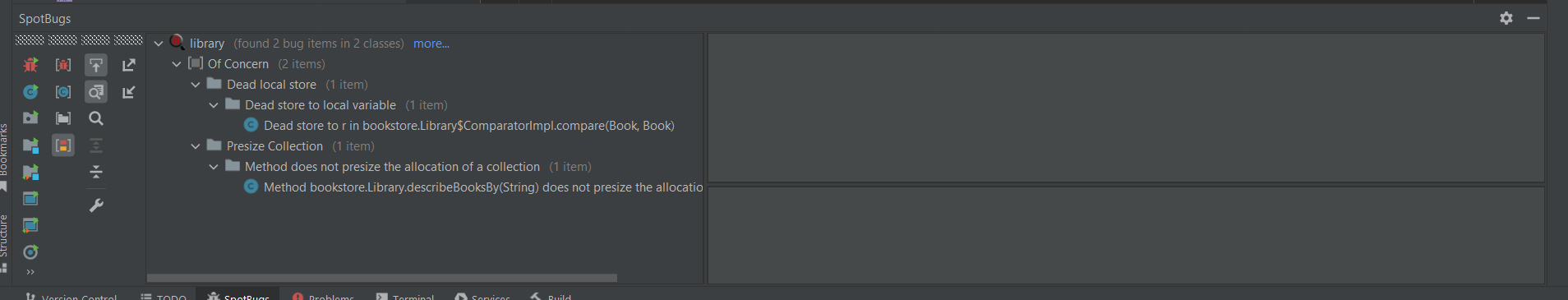


Рисунок 1 – работа SpotBugs с проектом library.

StopBugs успешно проверил данный проект, найдя несколько несущественных ошибок:

* Dead store to r;

Эта инструкция присваивает значение локальной переменной, но это значение не считывается и не используется ни в одной последующей инструкции. Часто это указывает на ошибку, поскольку вычисленное значение никогда не используется.

Компилятор Sun javac часто создает мертвые хранилища для конечных локальных переменных. Поскольку SpotBugs — это инструмент на основе байт-кода, не существует простого способа устранить эти ложные срабатывания.

* Method describeBooksBy(String) does not presize the allocation of a collection.

Метод не определяет размер коллекции.

Этот метод выделяет коллекцию с помощью конструктора по умолчанию, даже если априори известно (или, по крайней мере, можно разумно предположить), сколько элементов будет помещено в коллекцию, и, таким образом, без необходимости вызывает промежуточные перераспределения коллекции.

Возможно, использовать конструктор, который принимает начальный размер, и это будет намного лучше, но из-за loadFactor карт и наборов даже это не будет правильной оценкой.

Если использовать Guava, его методы, которые выделяют карты и наборы с заранее определенным размером, чтобы получить наилучшие шансы на отсутствие перераспределения, например:

Sets.newHashSetWithExpectedSize (целое число)

Maps.newHashMapWithExpectedSize(int)

Если нет, хорошей оценкой будет ожидаемый размер / {LOADING\_FACTOR}, который по умолчанию равен 0,75.

# 3 Анализ полученных результатов

В данной лабораторной работе была осуществлена проверка проекта library на наличие проблем у проекта, были обнаружены незначительные ошибки в коде.

Также была подключения библиотеки jsr305-2.0.0.jar.

Данная работа позволила изучить работу с анализаторами кода и применить знания на практических примерах.