Министерство науки и высшего образования РФ

ФГАОУ ВО «Уральский федеральный университет

имени первого Президента России Б. Н. Ельцина»

ИРИТ-РТФ

Центр ускоренного обучения

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ N 4**

по дисциплине «Конструирование программного обеспечения»

**Тема:** Статический анализ кода

|  |  |
| --- | --- |
| Студент группы РИЗ-490028у: | А.А. Трифонов  Д.А. Житников  Д.В. Федяков  В.Н. Жирнова |
| Преподаватель: | С.И. Тимошенко,  доц., к.т.н |

Екатеринбург 2022

**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.

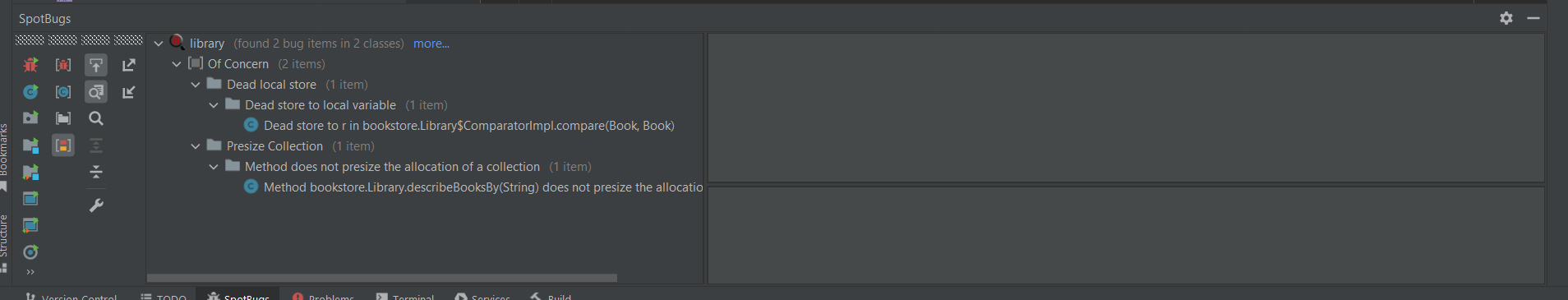


Рисунок 1 – работа SpotBugs с проектом library.

FindBugs 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 результатов**

В данной 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.