**ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА АНАЛИЗА КОДА ПРОГРАММ**

# **Исследование программы с использованием статического анализатора Cppcheck**

Используя статический анализатор, удалось выяснить, что в программе присутствует одна ошибка производительности. Результаты работы анализатора представлены на рисунке 1.

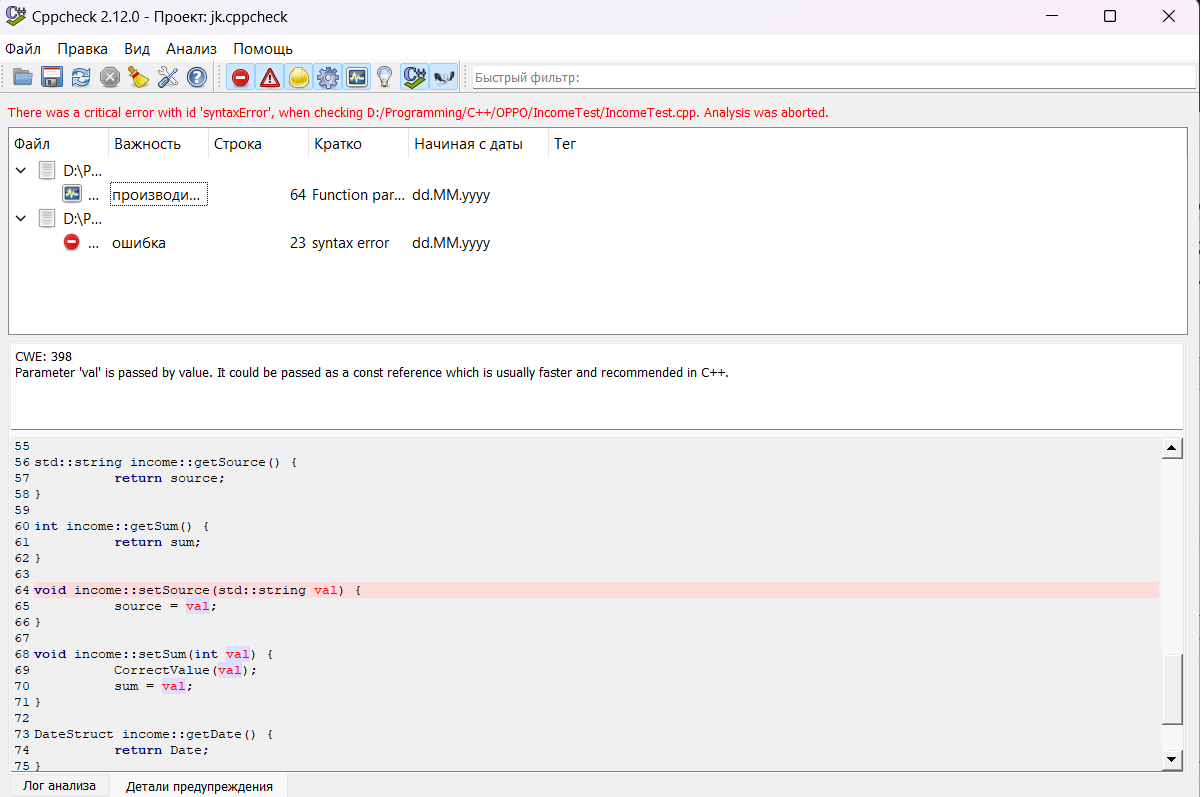


Рисунок 1 – результат работы анализатора Cppcheck.

Для исправления ошибки типа “CWE398” было принято решение об использовании ссылок при передаче аргументов в функцию. Исправление ошибки представлено на рисунке 2.

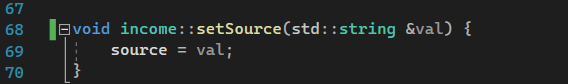


Рисунок 2 – Исправление ошибки “CWE398”

# **Исследование программы с использованием статического анализатора PVS-Studio**

Используя анализатор PVS-Studio удалось выявить несколько ошибок, разделённых различными категориями. Ошибки вида “High”, найденные в программе, представлены на рисунке 3.

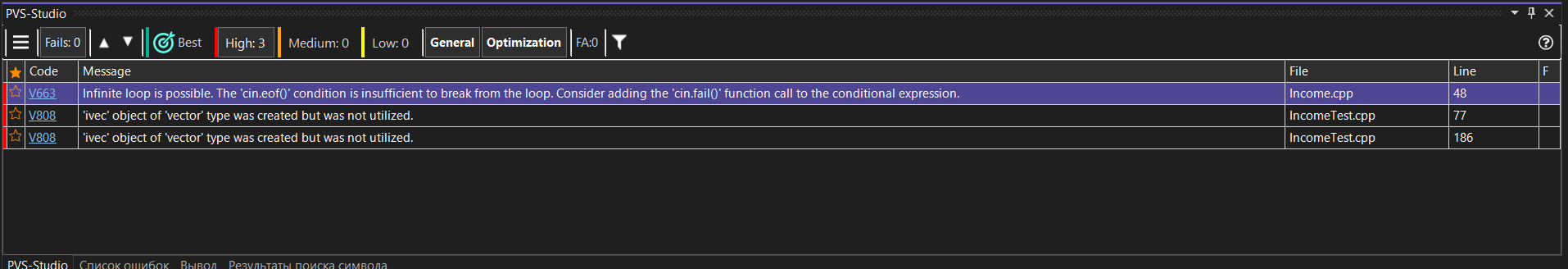


Рисунок 3 – Ошибки вида “High”, найденные в программе.

Как видно из рисунка 3, было найдено две ошибки в файлах: Income.cpp и incomeTest.cpp. Для исправления первой ошибки “*-V663*”было принято решение добавить в условие цикла while дополнительный параметр. Исправление первой ошибки представлено на рисунке 4.

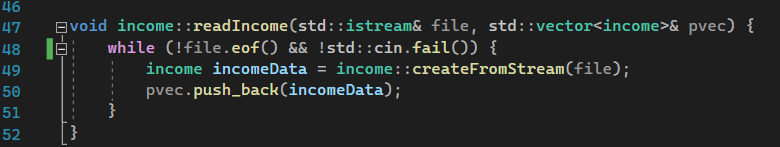


Рисунок 4 – Исправление ошибки V663.

Как видно из рисунка 4, в условие цикла было добавлено выражение “!std::cin.fail()”, что позволило корректно завершать цикл в случае, если чтение данных из файла будет неудачным.

Для исправления второй ошибки “*V808”* было принято решение об удалении лишних неиспользуемых векторов в файле incomeTest.cpp в функциях validIncomePrintTest() и invalidIncomePrintTest(). Исправление второй ошибки представлено на рисунке 5.

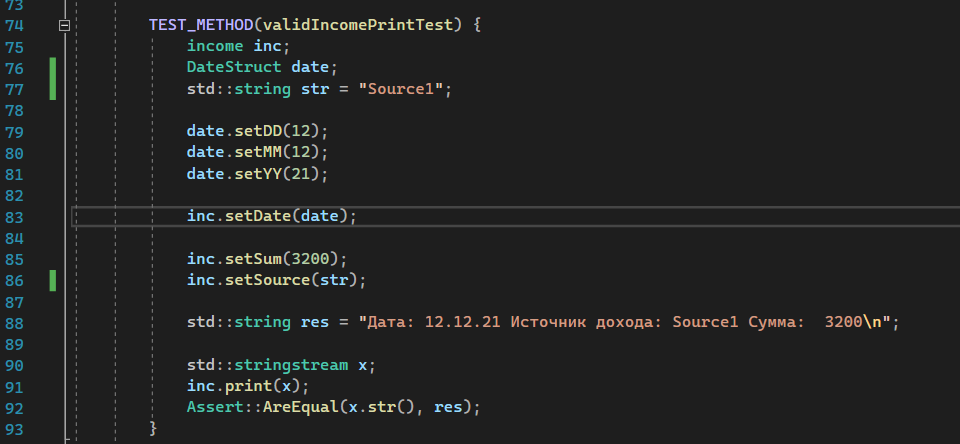


Рисунок 5 – Исправление ошибки V808

Аналогичные действия были произведены и со второй функцией.

Помимо ошибок вида “High”, были найдены также ошибки вида “Low”, которые являются незначительными. На рисунке 6 представлены найденные ошибки.

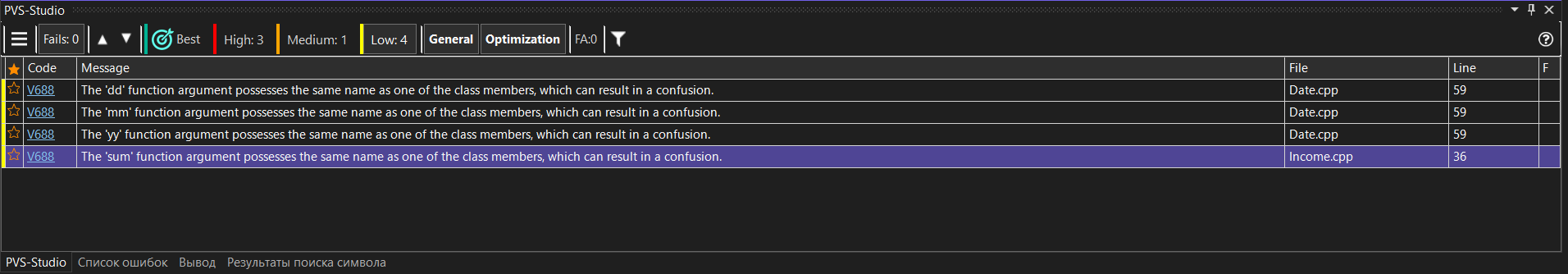


Рисунок 6 – Ошибки вида “Low”, найденные в программе.

Как видно из рисунка, в трёх файлах (Date.cpp, Income.cpp), были найдены ошибки одного вида – V688. Для решения ошибки V688, было принято решение о простом переименовании параметров функции Correct(). Исправление ошибки представлено на рисунке 7.

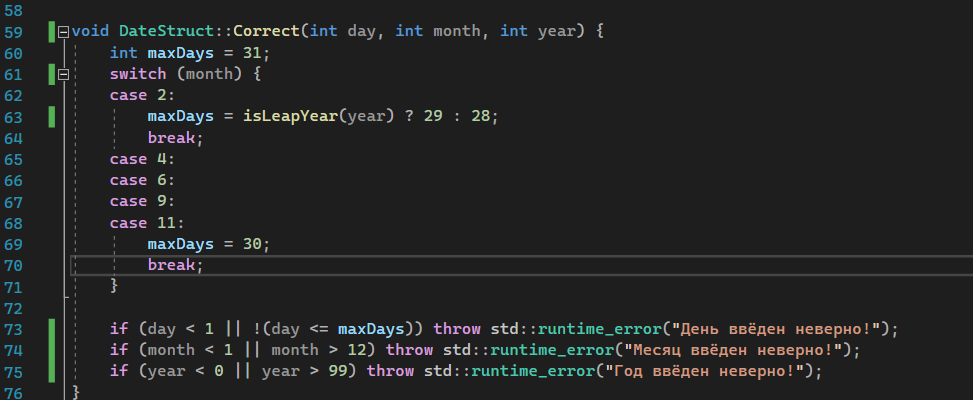


Рисунок 7 – Исправление ошибки V688.

Как видно из рисунка, старые параметры функции Correct() – dd, mm, yy, были заменены на более осмысленные day, month, year, что помогло решить ошибку.

Для исправления второй ошибки V688 было принято аналогичное решение о переименовании параметра функции CorrectValue(). Исправление ошибки V688 представлено на рисунке 8.

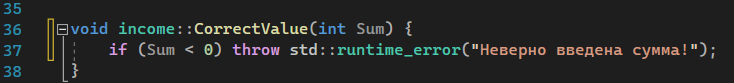


Рисунок 8 – Исправление ошибки V688.

Помимо всех вышеперечисленных ошибок, были найдены также ошибки вида “Medium”. На рисунке 9 представлены найденные ошибки.

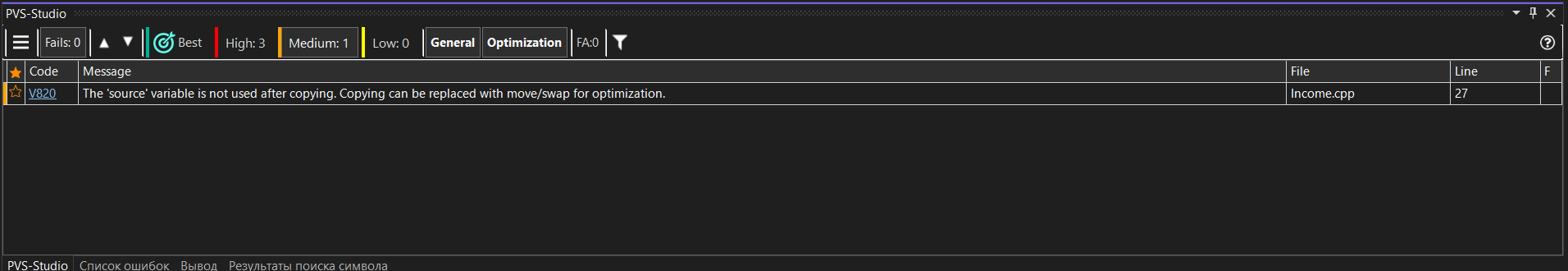


Рисунок 9 – Ошибки вида “Medium”, найденные в программе.

Как видно из рисунка 10, была найдена ошибка типа V820. На рисунке 10 представлено исправление этой ошибки.

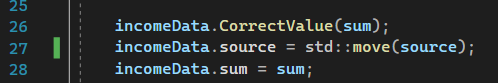


Рисунок 10 – Исправление ошибки V808.

Как видно из рисунка 10, для устранения ошибки было принято решение об использовании std::move в качестве функции, позволяющей передать значение source в incomeData.source, что помогло исправить ошибку.

После выполнения всех вышеперечисленных ошибок, анализатор уведомил о том, что ошибок больше в программе нет. Уведомление представлено на рисунке 11.

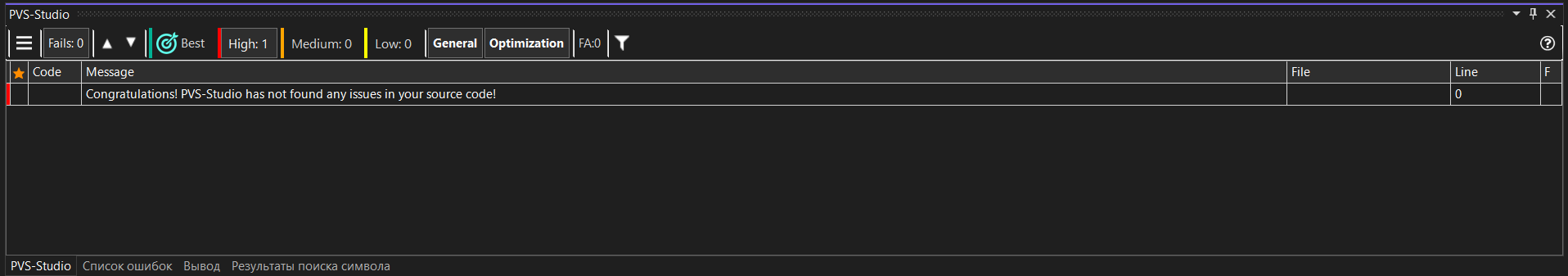


Рисунок 11 – Уведомление об отсутствии ошибок.

# **Исследование программы на возможные утечки памяти с использованием Visual Leak Detector**

Для проверки программы на всевозможные утечки памяти было принято решение об использовании утилиты Visual Leak Detector. Результат работы утилиты по умолчанию выводится в консоль. Для обеспечения её работоспособности необходимо подключить её, используя директиву #include

После подключения достаточно лишь запустить программу для проверки её на возможные утечки. Уведомление о том, что утечки памяти в программе отсутствуют, изображено на рисунке 12.

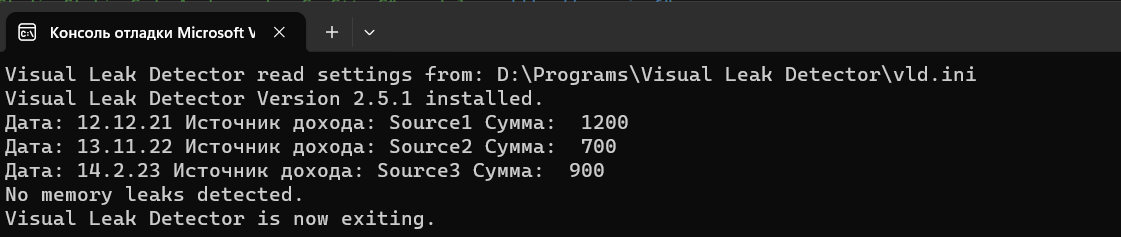


Рисунок 12 – Уведомление об отсутствии утечек памяти в программе.

Как видно из рисунка 12, в программе утечек памяти не наблюдается.