Федеральное государственное автономное образовательное учреждение

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий Кафедра вычислительной техники

**ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА АНАЛИЗА КОДА ПРОГРАММ**

Вариант №12

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.С. Васильев

подпись, дата

Студент Ки22-08Б, 032212205 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.Д. Федорова

подпись, дата

Красноярск 2023

**Содержание**

[1. Статический анализатор 4](#_Toc154089423)

[2. Динамический анализатор 7](#_Toc154089424)

[Заключение 8](#_Toc154089425)

**Введение**

Работа заключается в:

1. Исследовании программы полученной в ходе выполнения работы 3 с помощью инструментов статического и динамического анализа кода;

2. Оценке полученных решений в соответствии с критериями качества ПО;

3. улучшении программы - устранение обнаруженных недочетов.

# **Статический анализатор**

Для анализа использовался статический анализатор «Cppcheck». Найденные ошибки показаны ниже (Рисунок 1).

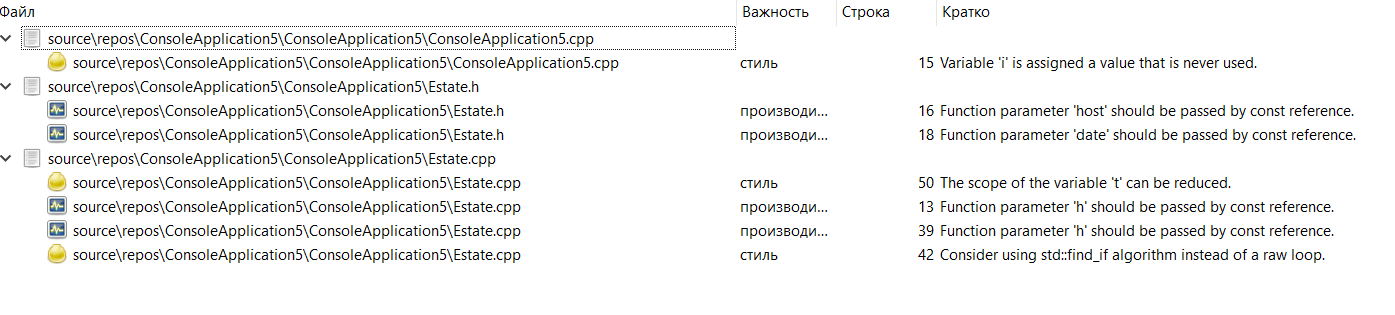
****

Рисунок 1 – ошибки найденные статическим анализатором

Первый дефект программы изображен на рисунке 2. Добавлена лишняя переменная i, которая не используется в дальнейшем коде.

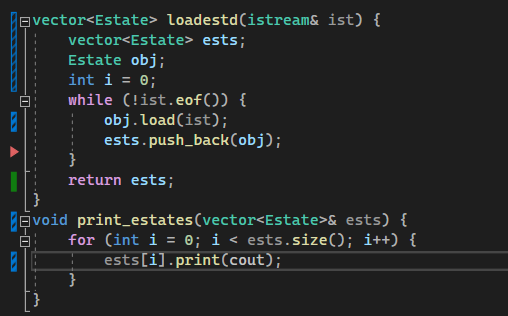
****

Рисунок 2 – первый дефект

Исправленный код заключается в удалении этой переменной – рисунок 3.

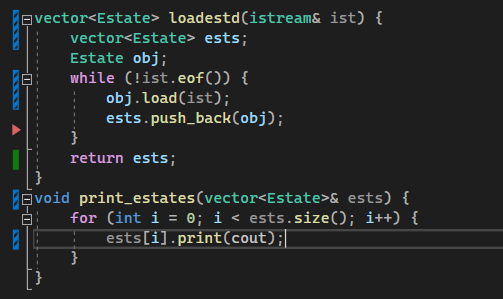
****

Рисунок 3 – исправленный код первого дефекта

Второй дефект программы изображен на рисунке 4. Ошибка заключается в лишнем присвоении переменной значения, передаваемого функции.

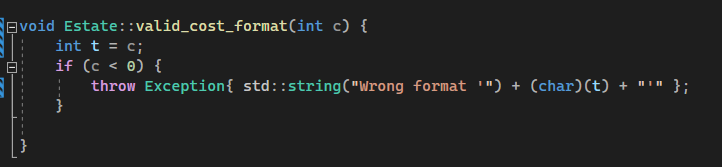
****

Рисунок 4 – второй дефект

Исправленный код не включает в себя эту переменную (Рисунок 5).

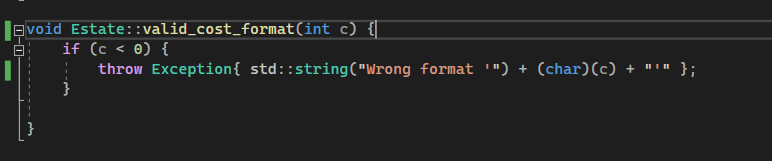
****

Рисунок 5 – исправленный код второго дефекта

Следующие ошибки относились к функциям valid\_host\_format, set\_host, set\_date, а также к конструктору класса Estate. Все ошибки заключались в неэффективном использовании памяти. Необходимо использовать константую ссылку. Исправленный код представлен на рисунках 6, 7, 8, 9, 10, 11.

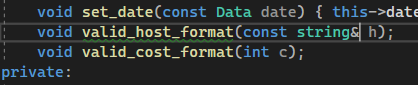
****

Рисунок 5 – исправленный код третьего дефекта в Estate.h

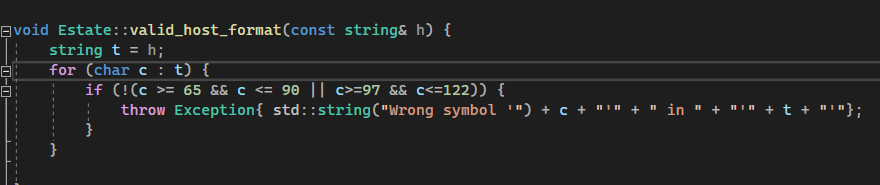
****

Рисунок 6 – исправленный код третьего дефекта в Estate.cpp

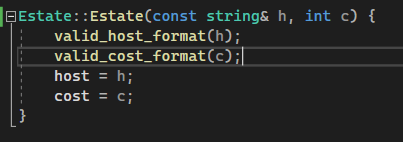
****

Рисунок 7 – исправленный код четвертого дефекта в Estate.cpp

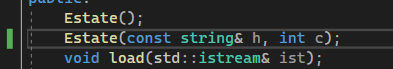
****

Рисунок 8 – исправленный код четвертого дефекта в Estate.h

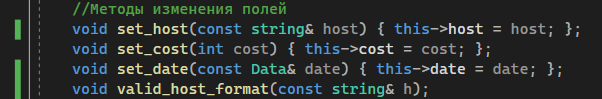
****

Рисунок 9 – исправленный код пятого дефекта в Estate.h

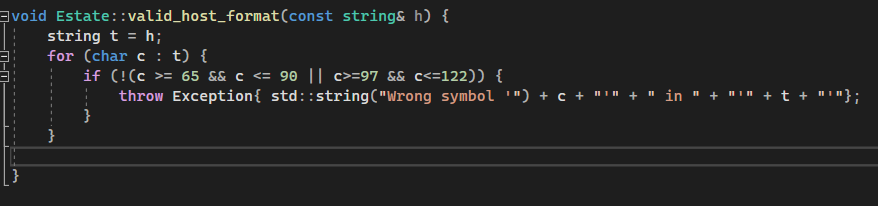
****

Рисунок 10 – исправленный код пятого дефекта в Estate.cpp

Также в «Cppcheck» было предложено исправить код функции проверки правильного формата записи строки в поле «host» в класс Estate. Функция, указанная на рисунке 10, была преобразована с помощью шаблонной функции find\_if\_not из библиотеки <algorithm>.

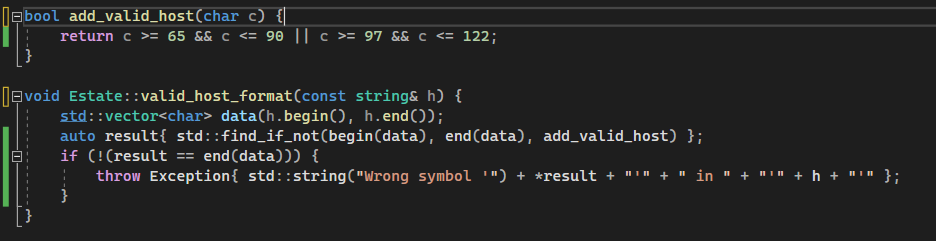
****

Рисунок 11 – исправленный код шестого дефекта в Estate.cpp

# **Динамический анализатор**

Для проверки утечек в памяти использовалась сторонняя библиотека, подключаемая к «Visual Studio» - «Visual Leak Detector». В ходе использования анализатора утечек в памяти не было обнаружено.

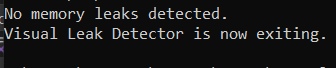


Рисунок 12 – результат работы динамического анализатора

# **Заключение**

В результате анализа было выявлено 7 дефектов программы с помощью статического анализатора «Cppcheck» и не было выявлено утечек памяти в ходе работы динамического анализатора «Visual Leak Detector». Найденные ошибки были исправлены в коде.