|  |
| --- |
| Министерство науки и высшего образования РФ Федеральное государственное автономное образовательное учреждение высшего профессионального образования **«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»** |
| Институт космических и информационных технологий |
| Кафедра вычислительной техники |

**Лабораторная работа №5**

**«**Инструментальные средства анализа кода программ**»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Преподаватель |  |  |  |  |  | Л.С. Артемьев |
|  |  |  |  |  | подпись, дата |  |
| Студент |  | КИ22-06Б, 032213011 |  |  | К.Ю. Михайлова |
|  |  | номер групп, зачетной книжки |  | подпись, дата |  |

Красноярск 2023

**Техническое задание**

1. Исследование программы полученной в ходе выполнения работы 3 с помощью инструментов статического и динамического анализа кода.
2. Оценка полученных решений в соответствии с критериями качества ПО.
3. Улучшение программы – устранение обнаруженных недочётов.

Таблица 1 – Задание по варианту

|  |  |
| --- | --- |
| Вариант | Задание |
| 12 | Недвижимость: владелец (строка), дата постановки на учет, ориентировочная стоимость(целое) |

**Ход работы**

Для выполнения 5 лабораторной работы мне потребовалось установить приложение «Cppcheck». Суть его работы заключается в анализе исходного кода, также после анализа программа выведет все твои недостатки и опишет проблему.

После анализа моего кода, программа показала следующее:

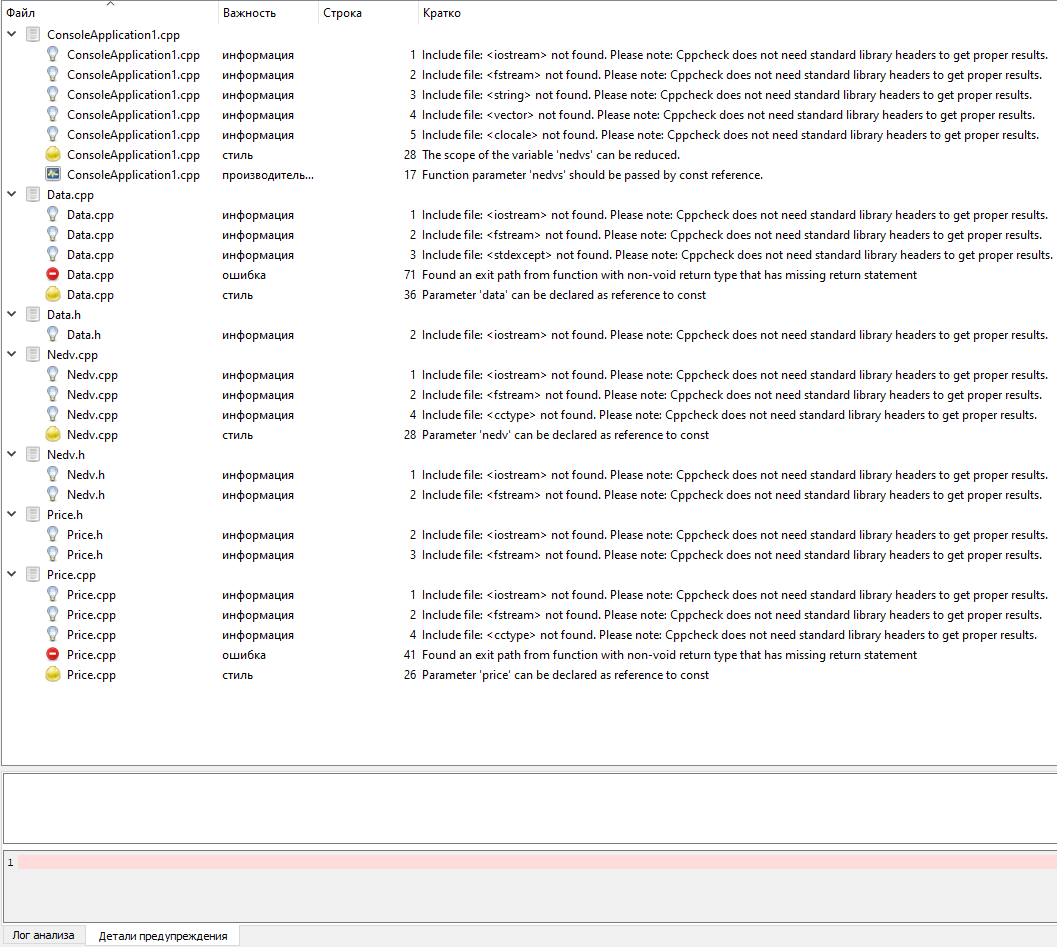


Рисунок 1 – Анализ программы

**Примечания:**

* Во вкладке «Файл» содержатся все открытые пользователем файлы;
* Вкладка «Важность» информирует пользователя о степени ошибочности:

1. Информация – информационная важность показывает на то, что библиотеки, содержащиеся в файлах не нужны программе для чтения;
2. Ошибка – программа обнаружила ошибку в загруженных файлах, указала, что за ошибка обнаружена и предлагает свой вариант решения это проблемы;
3. Производительность – показывает на то, как можно передать параметр по значению;
4. Стиль – показывает, как может быть объявлен параметр.

* Вкладка «Строка» показывает строку в файле, где была обнаружена ошибка;
* Вкладка «Кратко» содержит общую информацию об ошибочности конкретной строки в конкретном файле;

**Устранение полученных недочетов**

1. Чтобы устранить в «важности» ошибку в файле «Data.cpp» нужно написать return для пути выхода из функции с типом возврата non-void:

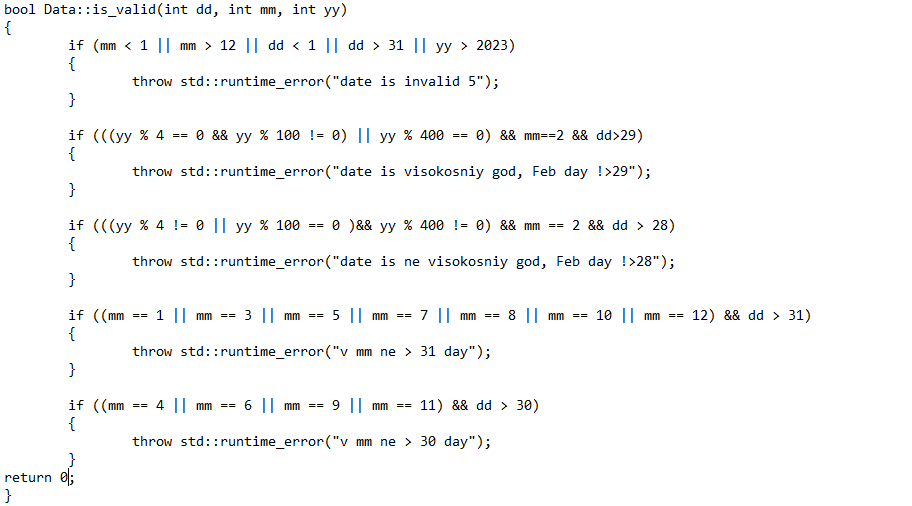


Рисунок 2 – Устранение ошибки

После написания данной строчки, анализ программы не найдёт эту ошибку:

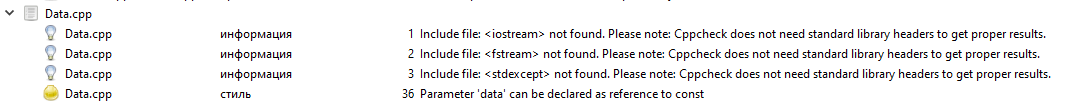


Рисунок 3 – Повторный анализ программы

1. Для устранения в «важности» ошибку, находящиеся в файле «Price.cpp» нам нужно проделать туже самую работу, что и в предыдущем пункте:

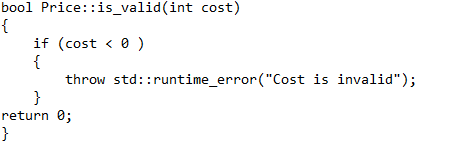


Рисунок 4 – Устранение ошибки

После написания строчки, повторный анализ программы не найдёт эту ошибку:

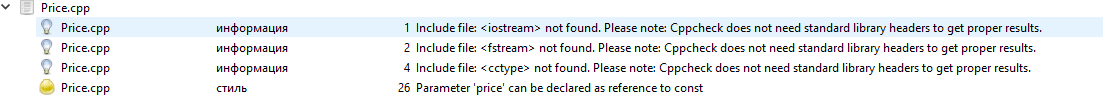


Рисунок 5 – Повторный анализ программы

1. Для устранения следующей «важности», а точнее «производительность» нам нужно передать параметр «nedvs», его нужно передать как ссылку const, что обычно быстрее, следовательно производительнее:

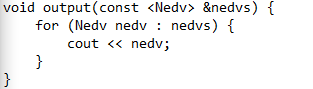


Рисунок 6 – Устранение ошибки

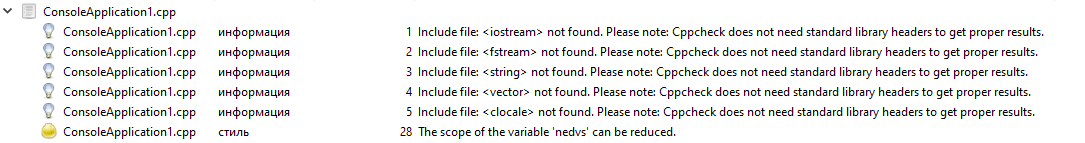


Рисунок 7 – Запуск анализа программы

1. Чтобы устранить ошибку «стиль» в файле «ConsoleApplication1.cpp» нужно предать «nedvs» не как «vector», а как «const»

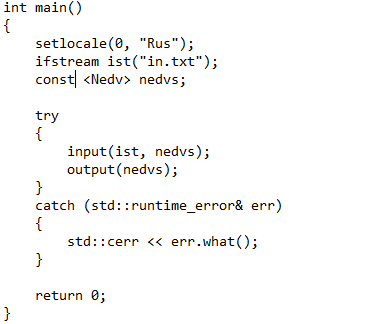


Рисунок 8 – Устранение ошибки

После запуска анализа ошибку будет устранена:

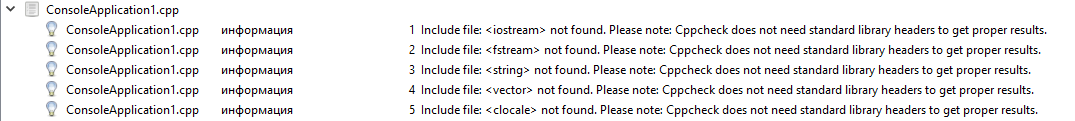


Рисунок 9 – Запуск анализа программы

1. В остальных «стилях» файлов: «Data.cpp», «Nedv.cpp» и «Price.cpp» обнаружена одна и таже ошибка: соответствующие параметры могут быть объявлены как ссылки на const:

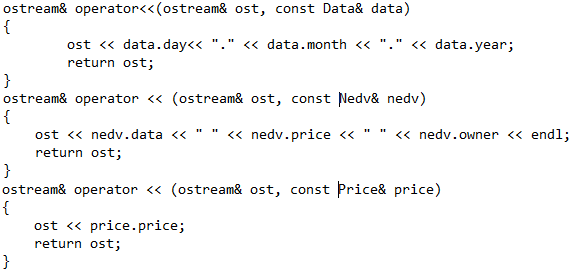


Рисунок 10 – Устранение ошибки

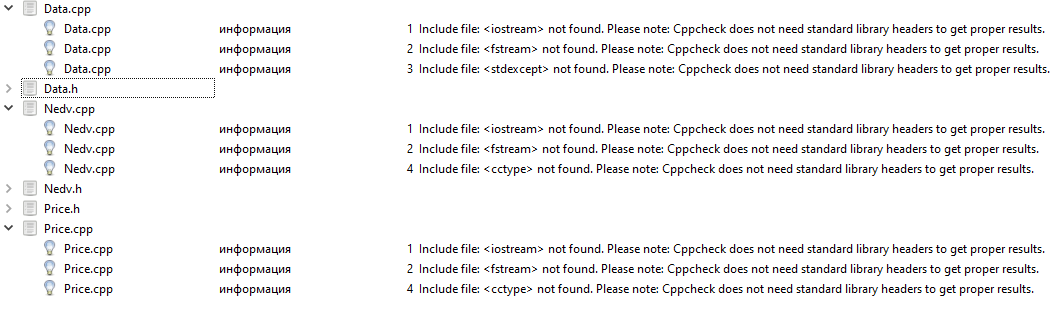


Рисунок 11 – Запуск анализа программы

1. Во вкладке важность – критерий информация во всех файлах содержит одну и туже проблему. Cppcheck не нуждается в заголовках стандартных библиотек для получения правильного результата. Для решения этих проблем, в каждом файле нужны удалить библиотеки.

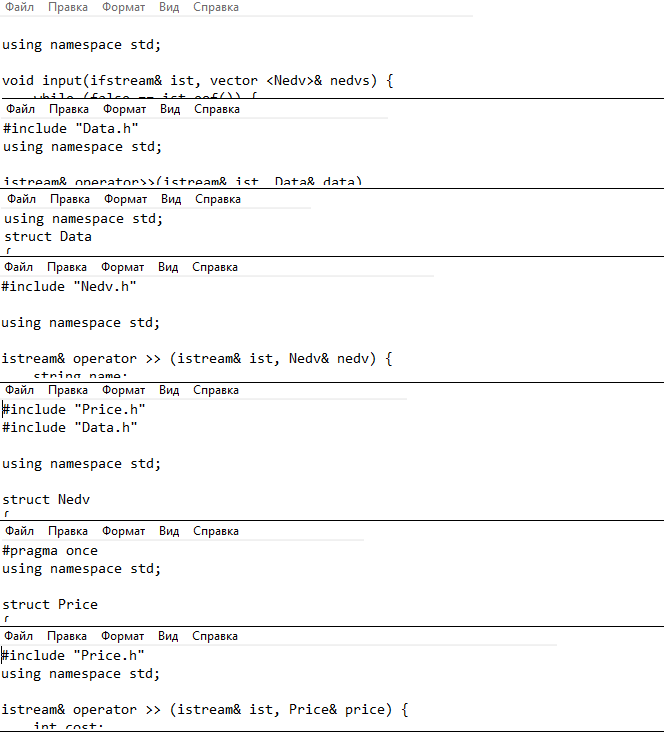


Рисунок 12 – Устранение ошибки

После устранения проблемы запускаем анализ программы:

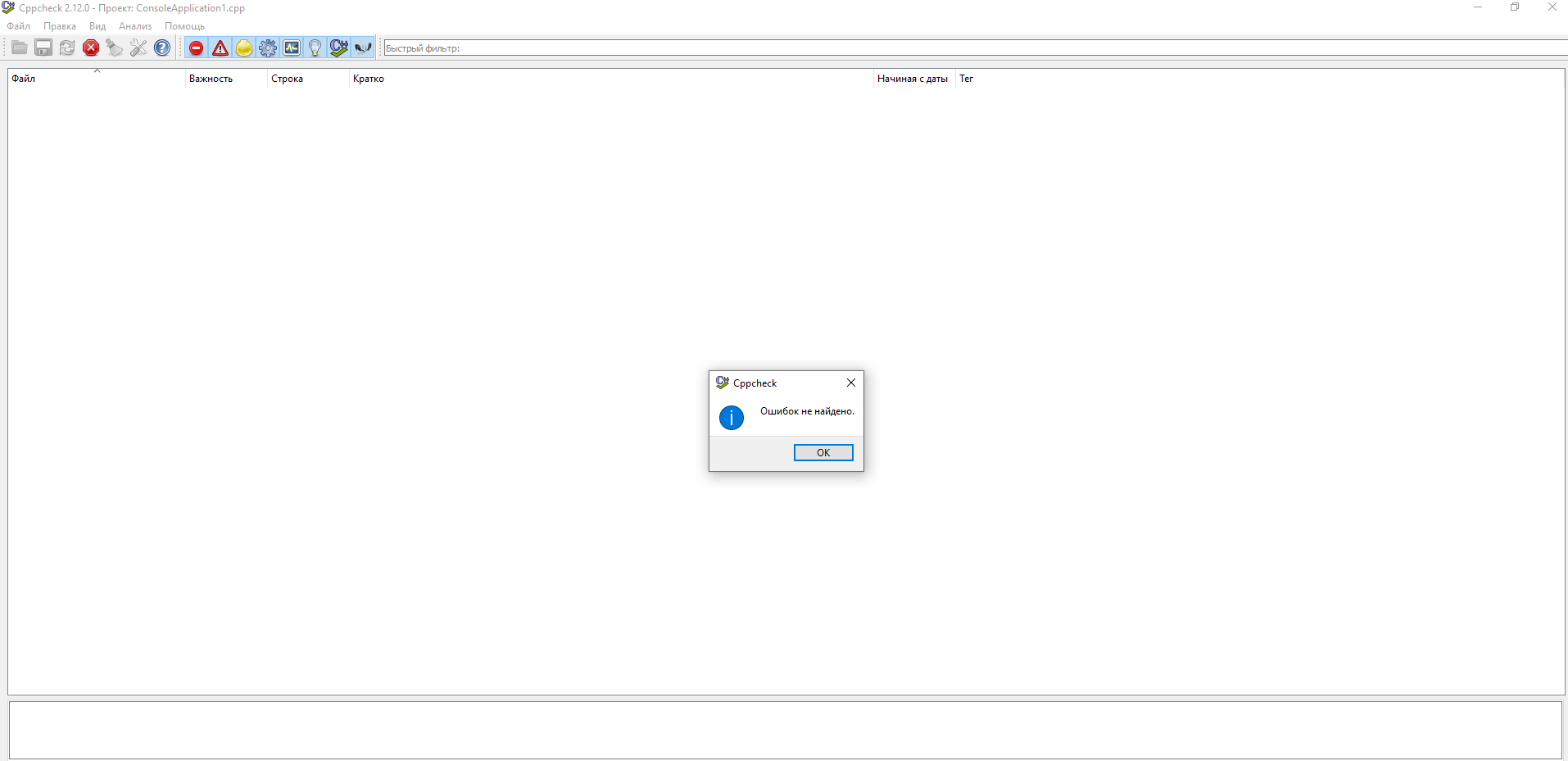


Рисунок 13 – Запуск анализа программы

**Устранение полученных недочетов**

1. Для применения динамического анализатора я взял пример утечки памяти. Для этого я использовал библиотек CRT. Для её подключения в начале основного файла «ConsoleApplication3» прописываем данные команды:

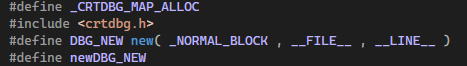


Рисунок 14 – Применение CRT

Также перед завершением программы прописываем: «\_CrtDumpMemoryLeaks ();».

После запуска программы получим данные об утечки памяти:

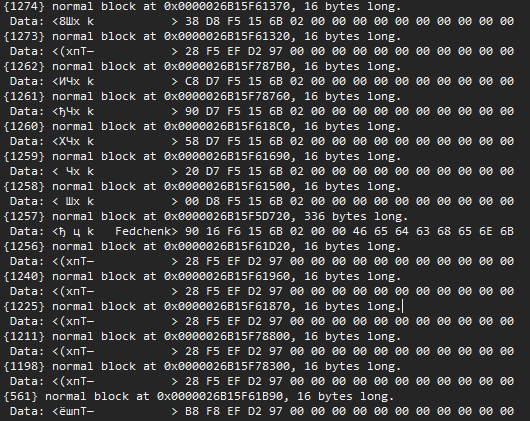


Рисунок 15 – Утечка памяти

**Заключение**

В ходе выполнения лабораторной работы были получены новые и закреплены старые навыки анализа ошибок и их устранения, также получены знания в оценке полученных решений в соответствие с критериями качества ПО. Поставленная задача решена полностью. Её итогом стала программа, не имеющая ошибок.