|  |
| --- |
| 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 |

**Практическая работа №5**

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Преподаватель |  |  |  |  |  | Л.С. Артемьев |
|  |  |  |  |  | подпись, дата |  |
| Студент |  | КИ22-07Б, 032217285 |  |  | А.А. Евченко |
|  |  | номер групп, зачетной книжки |  | подпись, дата |  |

Красноярск 2024

**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.
4. Исследование программы с использованием статического анализатора

Исследование будет проводиться с использование статического анализатора PVS-Studio. После анализа нашего кода получаем результат, показанный на рисунке 1.

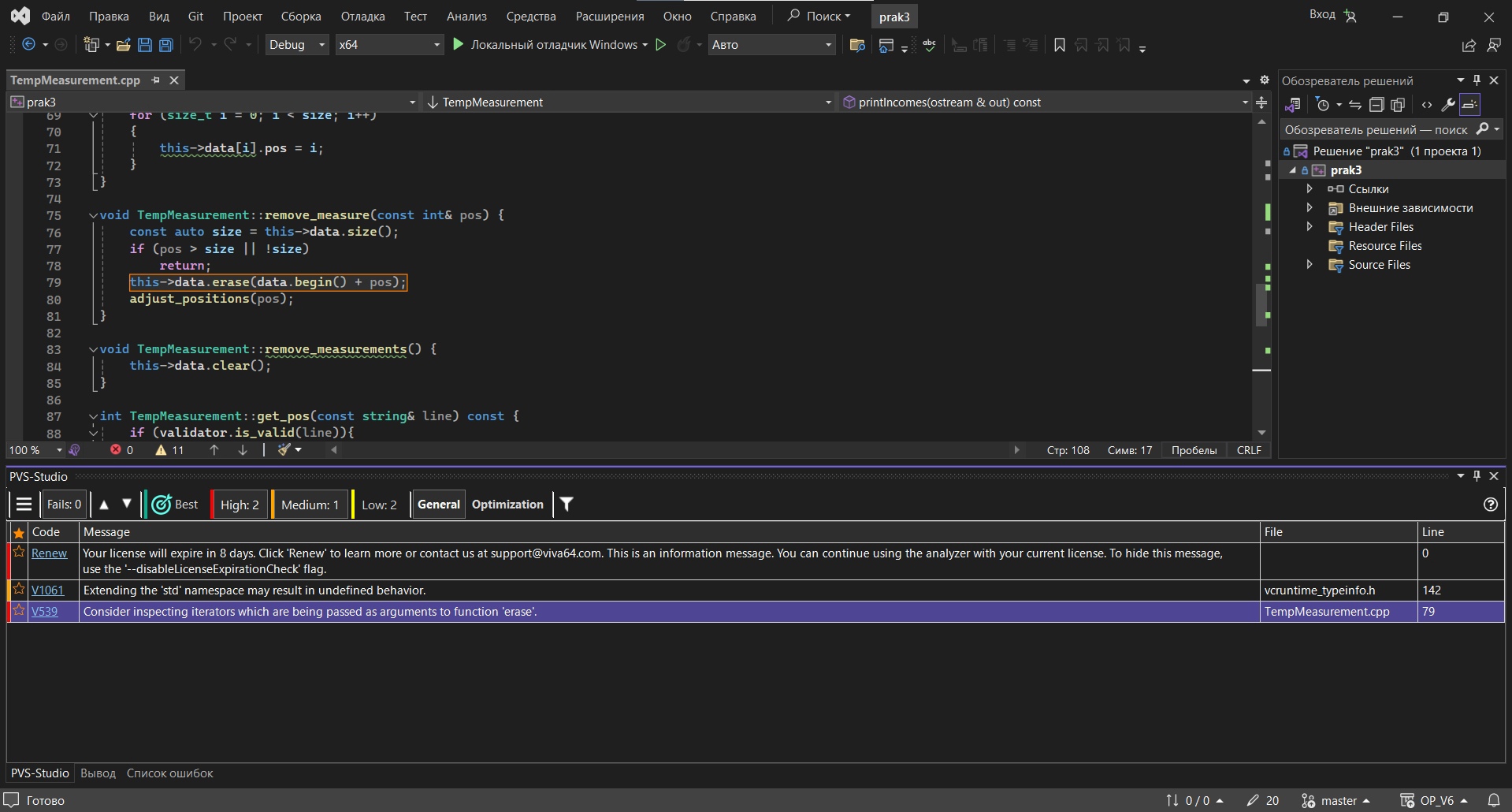


Рисунок 1 – Результат проверки в PVS-Studio

**Устранение полученных недочетов**

Ошибка V539 указывает на потенциальные проблемы с использованием итераторов, передаваемых в функцию erase. В данном случае ошибка связана с строкой кода:

“ **this->data.erase(data.begin() + pos);** ”

Чтобы устранить эту ошибку, я предлагаю явно указать пространство имен для метода erase. Нужно заменить эту строку кода на:

“ **this->data.erase(this->data.begin() + pos);** “

Такая замена позволит избежать неоднозначности и может решить проблему, связанную с V539.

После устранения всех ошибок запускаем анализатор кода опять и видим:

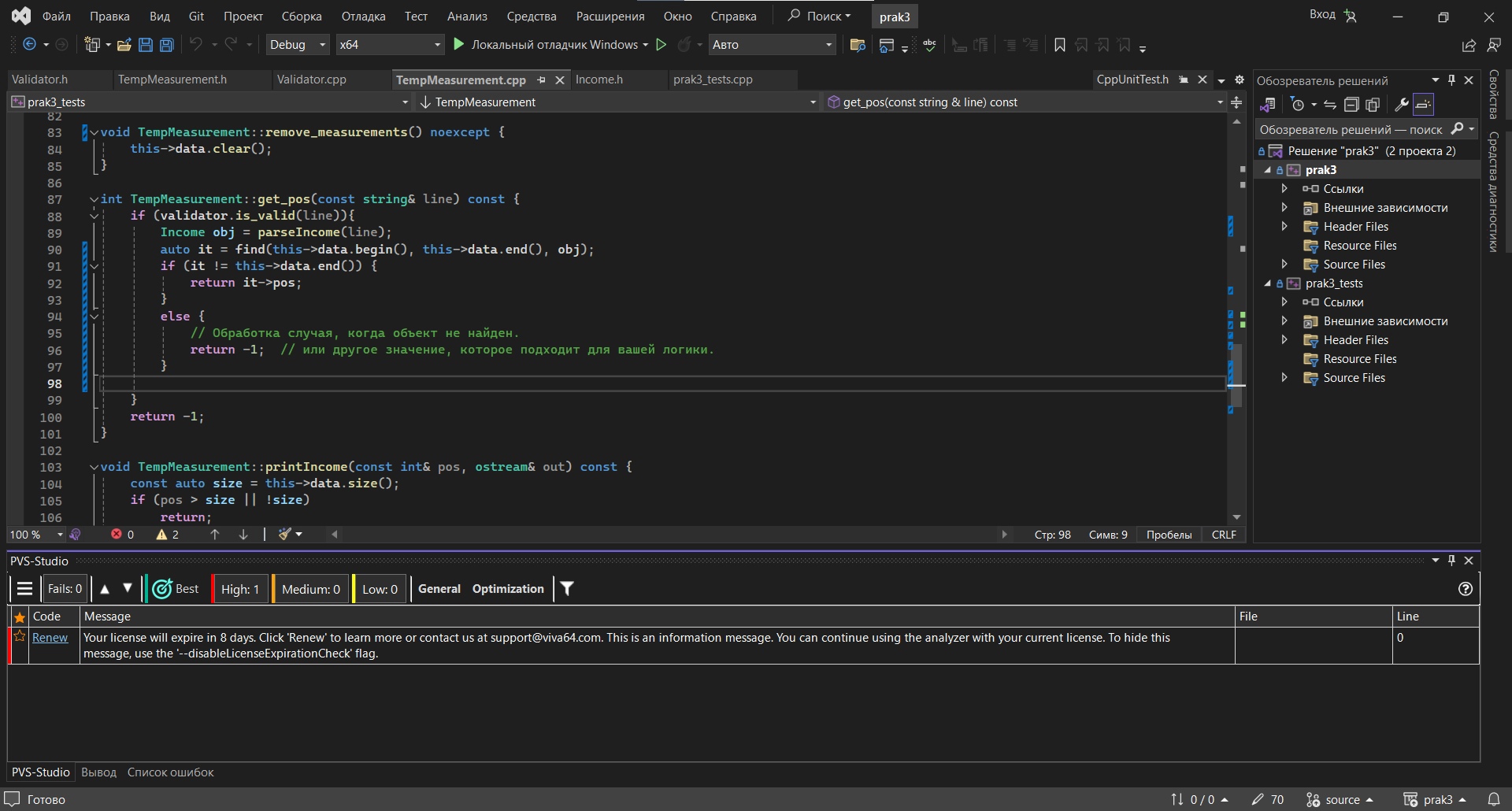


Рисунок 2 – Исправленный вариант кода.

Оставшиеся ошибки указывают на то, что у меня осталось 8 дней пробной версии анализатора кода. Соответственно все ошибки по коду были исправлены.

1. Исследование программы с использование динамического анализатора

Воспользуемся динамическим анализатором CRT для проверки на утечки памяти. Для этого добавим макросы в самое начало в основном файле, а в конце работы программы вызовем функцию \_CrtDumpMemoryLeaks().

Изменения можно посмотреть в фиксации с номером fb06e8d4.

После выполнения программы не получаем сообщений о найденных утечках, а значит программа не имеет утечек памяти.

**Заключение**

В ходе выполнения практической работы я воспользовался статическим анализатором PVS-Studio который помог мне обнаружить ошибки, которые, впоследствии, я исправил в процессе выполнения работы. Также я воспользовался динамическим анализатором, который помогает отследить утечки памяти, в данном случае утечек памяти выявлено не было. Поставленное изначально задание было выполнено.