

## Статический анализатор:

В качестве статического анализатора использовался плагин PVS-Studio. Анализатор показал одну ошибку (Рисунок 1).

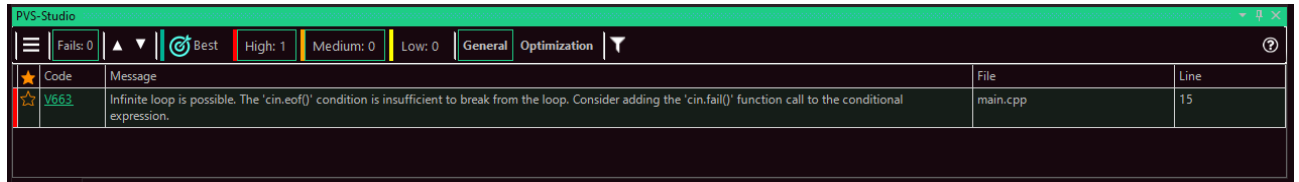


Рисунок 1 – Ошибки PSV-Studio

На Рисунке 2 код до исправления ошибки «Infinity loop...».



Рисунок 2 – Код до исправления

На Рисунке 3 код после исправления (добавлена дополнительная проверка «...in.good()...»).



```
4  #include <iostream>
5  #include <vector>
6  #include <fstream>
7  #include <locale>
8  #include <windows.h>
9
10 #include "cars.h"
11 #include "date.h"
12 using namespace std;
13
14 void loadData(vector<Car>& cars, istream& in) {
15     while (!in.eof() && in.good()) {
16         Car c;
17         c.load(in);
18         cars.push_back(c);
19     }
20 }
21
22 void printData(vector<Car>& cars) {
23     for (auto& c : cars) {
24         c.print();
25     }
26 }
27
28 int main()
29 {
30     setlocale(LC_ALL, "Russian");
31     SetConsoleCP(1251);
32     SetConsoleOutputCP(1251);
33     vector<Car> cars;
34     ifstream in("ist.txt");
35     try {
36         loadData(cars, in);
37         printData(cars);
```

Рисунок 3 – Код после исправления

На Рисунке 4 изображена итоговая проверка, ошибок не обнаружено.

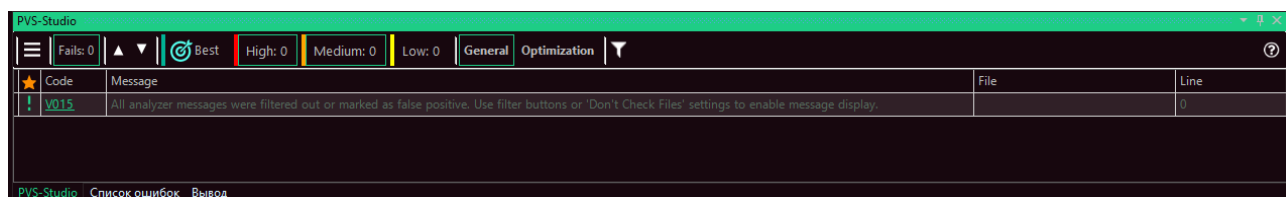
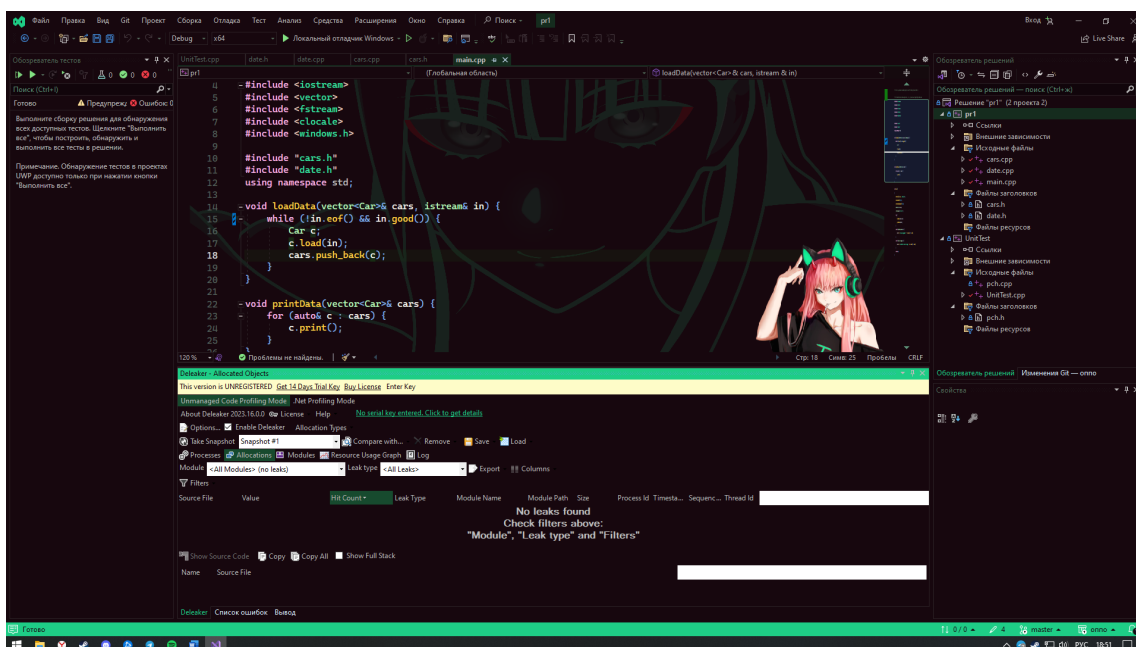


Рисунок 4 – Итоговая проверка

## Динамические анализаторы:

Использовалось два анализатора: Deleaker (аналог Valgrind) и библиотека «CRT». При использовании Deleaker утечек не обнаружено (Рисунок 5).



### Рисунок 5 – Вывод Deleaker

Первый запуск CRT показал утечки (Рисунок 6).

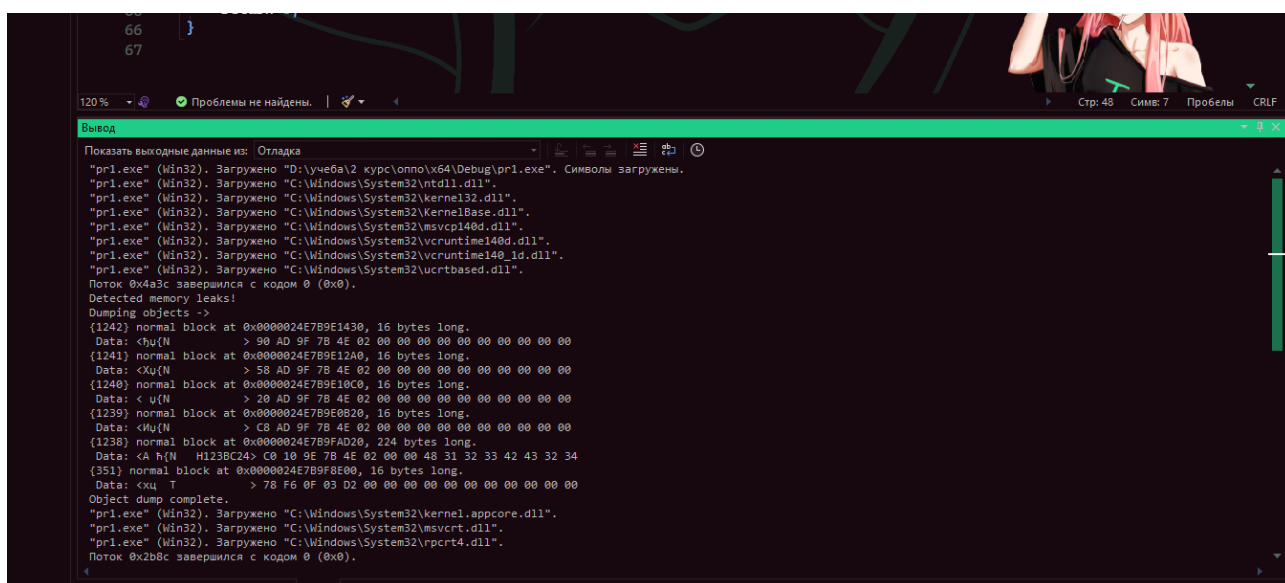
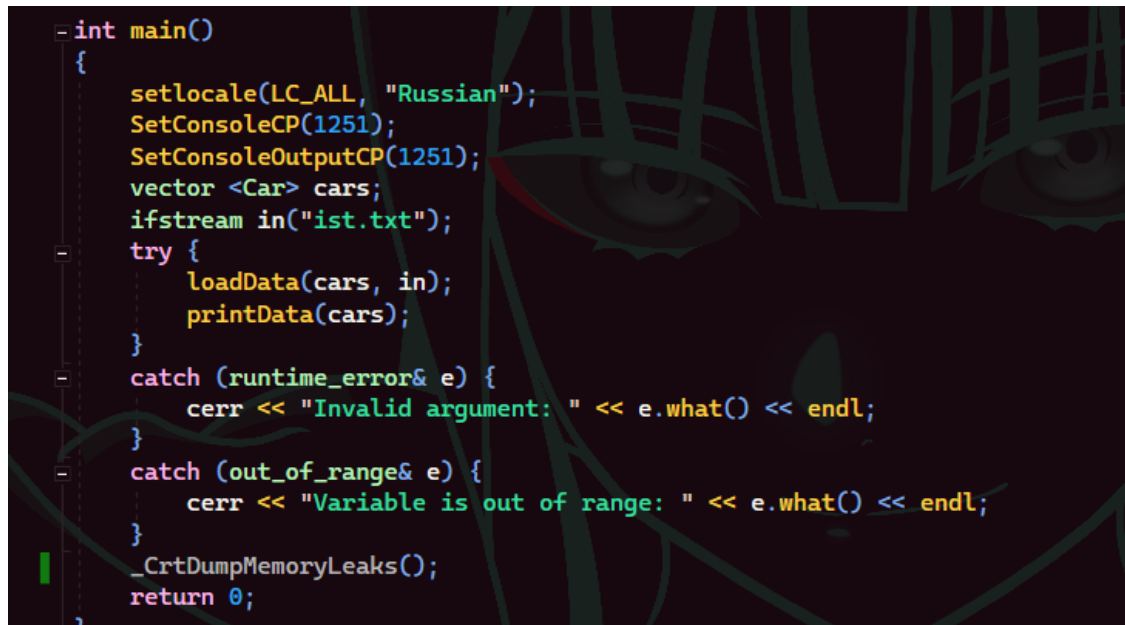


Рисунок 6 – Утечки обнаруженные CRT

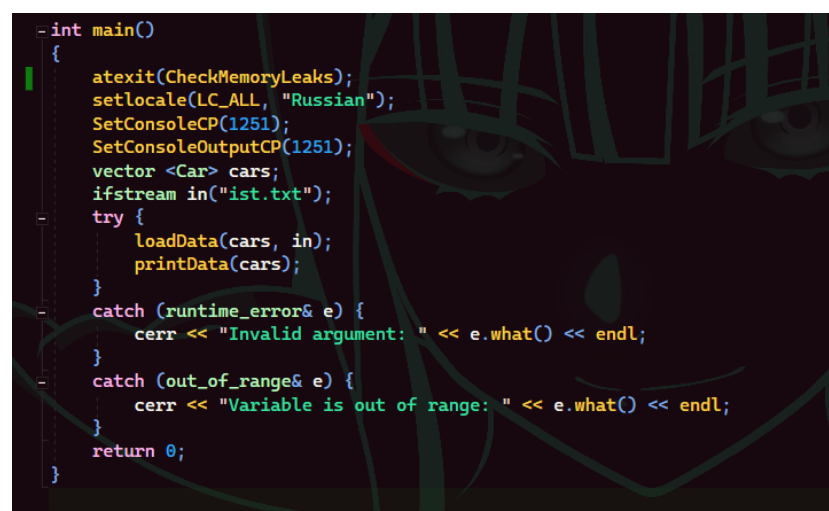
Утечки оказались не действительными. Они возникали из-за того, что функция `_CrtDumpMemoryLeaks()` вызывалась до выхода за пределы видимости `main` и `vector<Car> cars` не был очищен на момент её запуска. На Рисунке 7 код до исправления.



```
int main()
{
    setlocale(LC_ALL, "Russian");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    vector<Car> cars;
    ifstream in("ist.txt");
    try {
        loadData(cars, in);
        printData(cars);
    }
    catch (runtime_error& e) {
        cerr << "Invalid argument: " << e.what() << endl;
    }
    catch (out_of_range& e) {
        cerr << "Variable is out of range: " << e.what() << endl;
    }
    _CrtDumpMemoryLeaks();
    return 0;
}
```

Рисунок 7 – Код до исправления

На Рисунке 8 код после исправления.



```
int main()
{
    atexit(CheckMemoryLeaks);
    setlocale(LC_ALL, "Russian");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    vector<Car> cars;
    ifstream in("ist.txt");
    try {
        loadData(cars, in);
        printData(cars);
    }
    catch (runtime_error& e) {
        cerr << "Invalid argument: " << e.what() << endl;
    }
    catch (out_of_range& e) {
        cerr << "Variable is out of range: " << e.what() << endl;
    }
    return 0;
}
```

Рисунок 8 – Код после исправления

После этих исправлений никаких ошибок не было обнаружено.