

Статический анализатор:

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

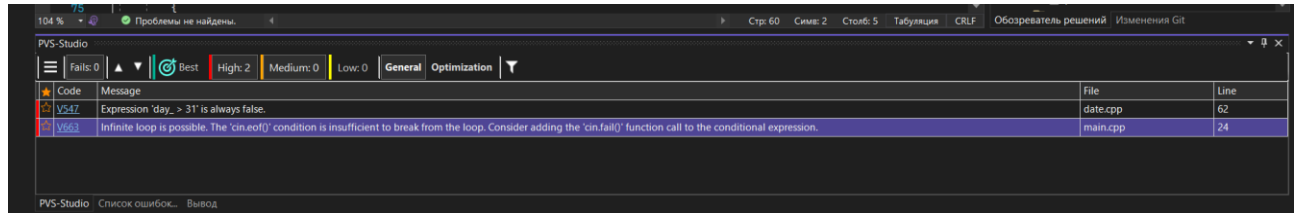


Рисунок 1 – Ошибки PSV-Studio

На Рисунке 2 код до исправления ошибки «Expression day > 31...».

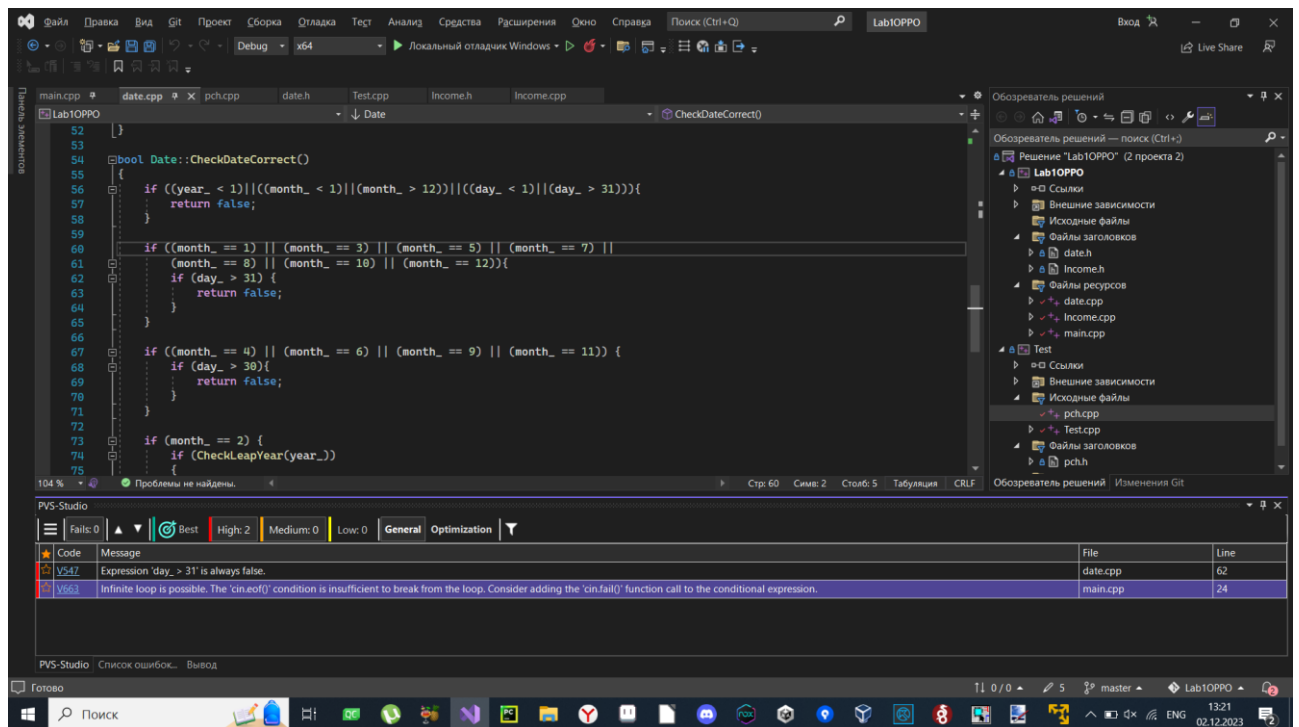


Рисунок 2 – Код до исправления

На рисунке 3 код после исправления (было удалено лишнее условие).

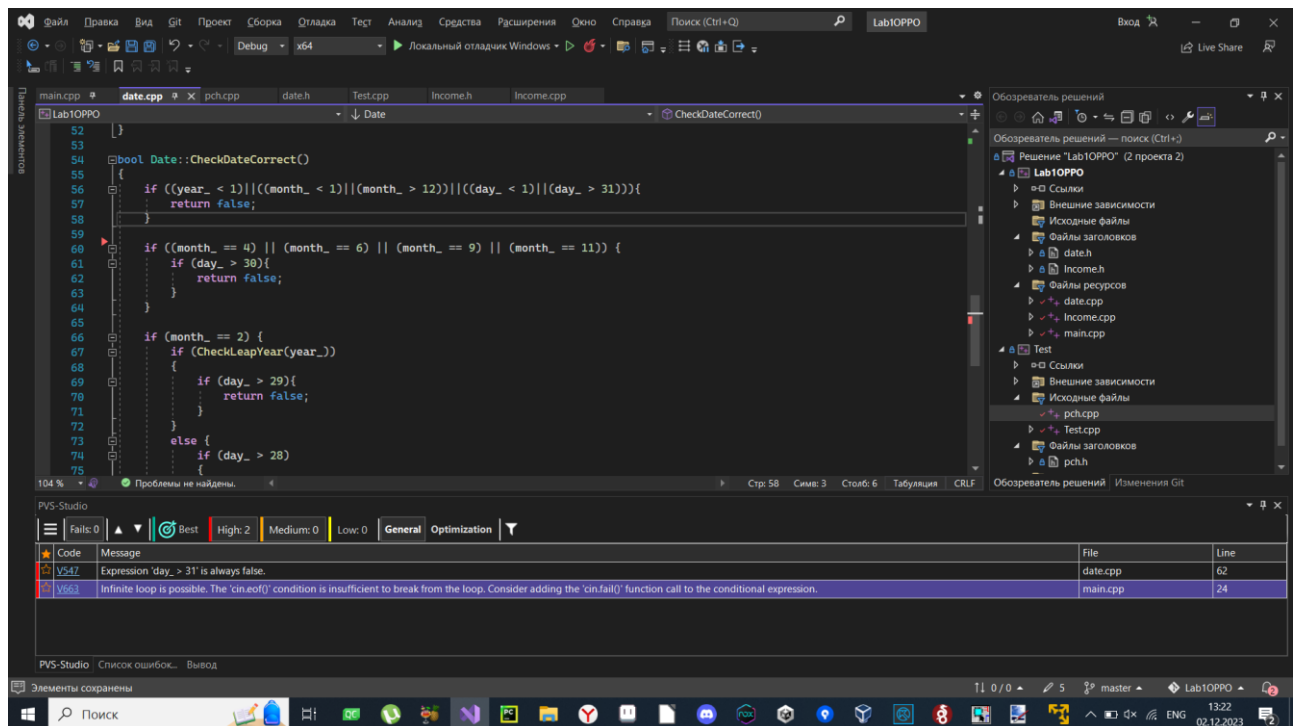


Рисунок 3 – Код после исправления

На Рисунке 4 код до исправления ошибки «Infinity loop...».

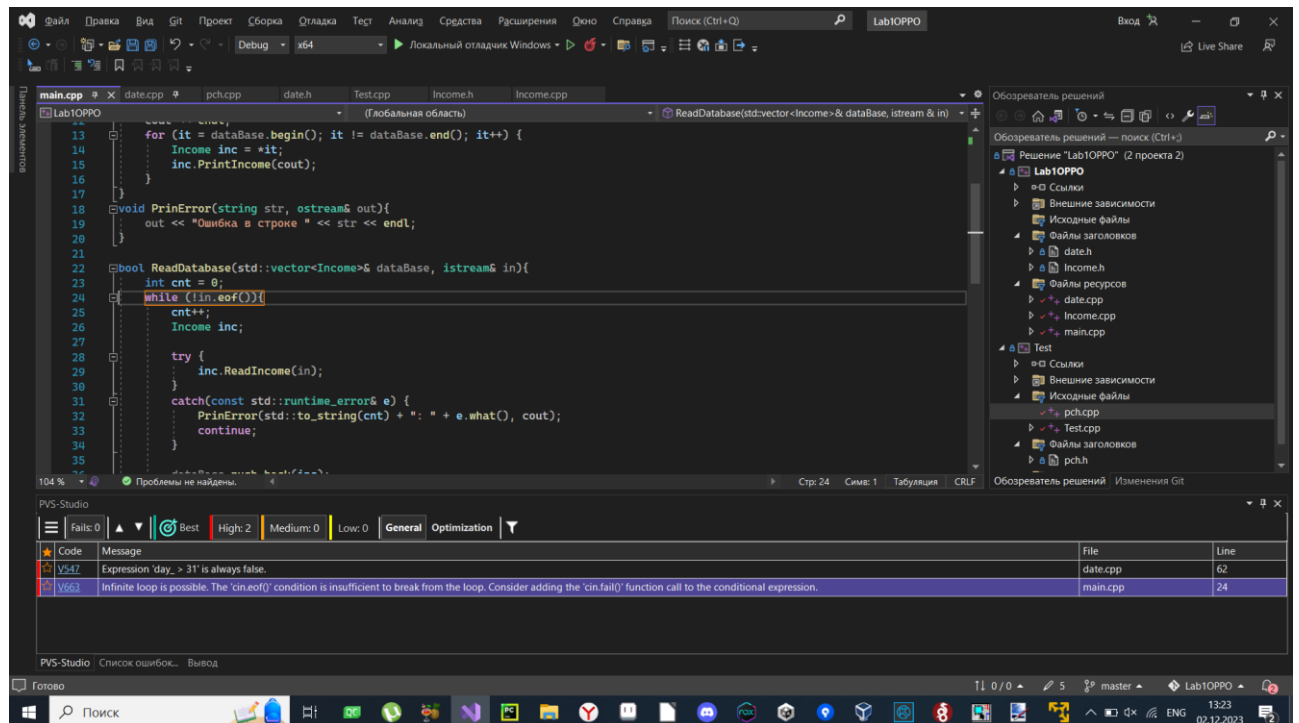


Рисунок 4 – Код до исправления

На Рисунке 5 код после исправления (добавлена дополнительная проверка «...in.good()...»).

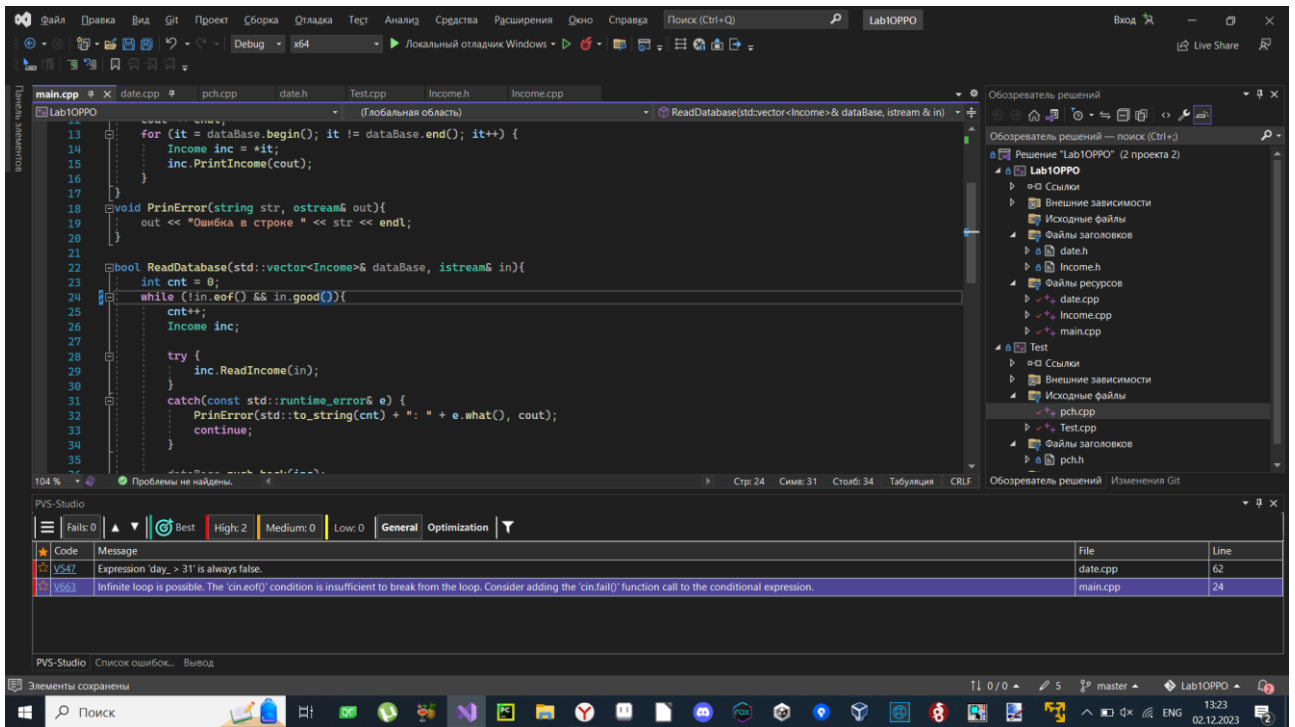


Рисунок 5 – Код после исправления

На Рисунке 6 изображена итоговая проверка, ошибок не обнаружено.

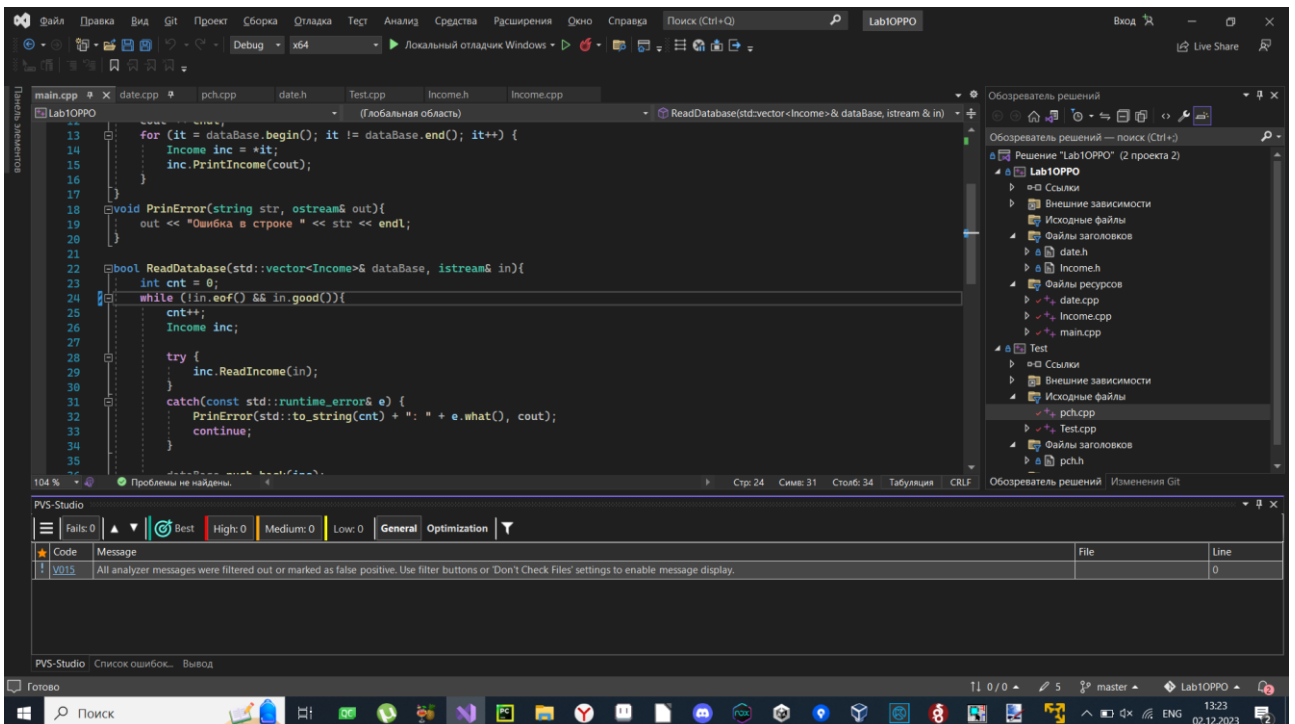


Рисунок 6 – Итоговая проверка

Динамические анализаторы:

Использовалось два анализатора: Deleaker (аналог Valgrind) и библиотека «CRT». При использовании Deleaker утечек не обнаружено (Рисунок 7).

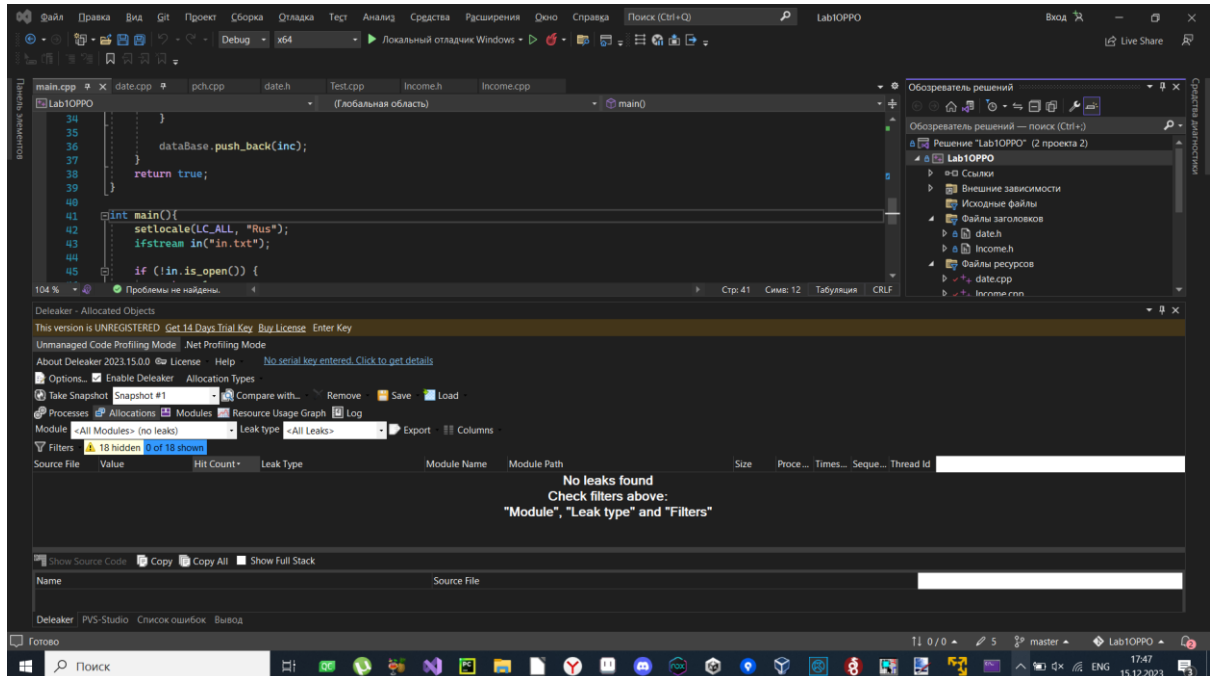


Рисунок 7 – Вывод Deleaker

Первый запуск CRT показал утечки (Рисунок 8).

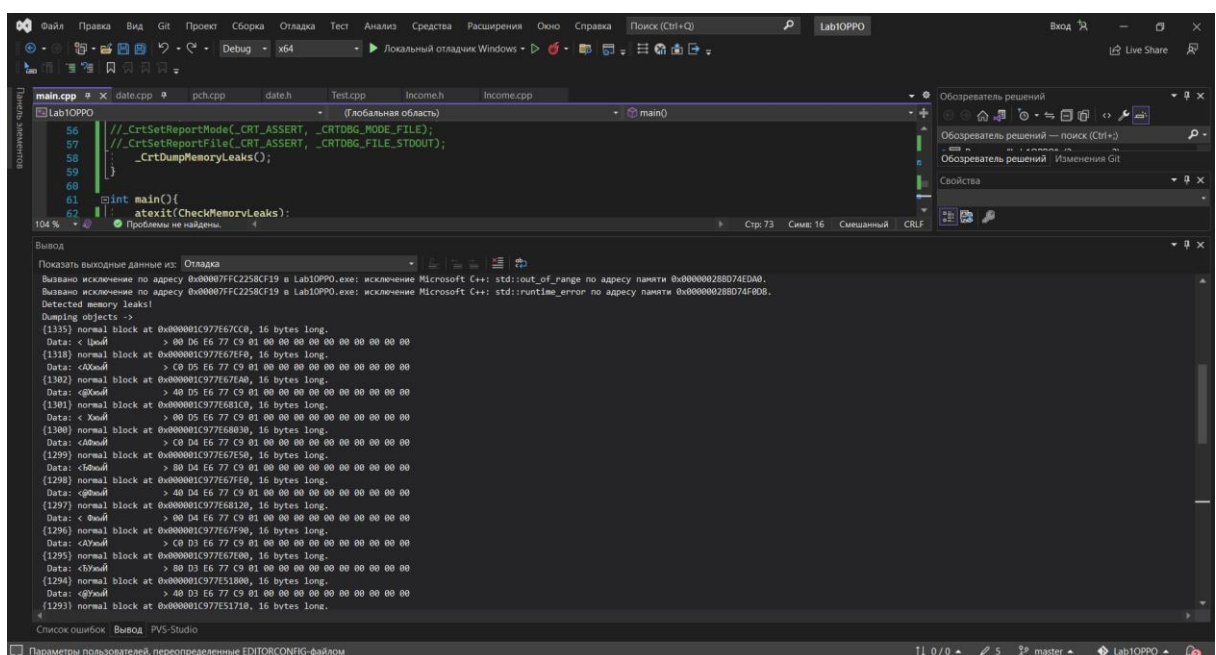
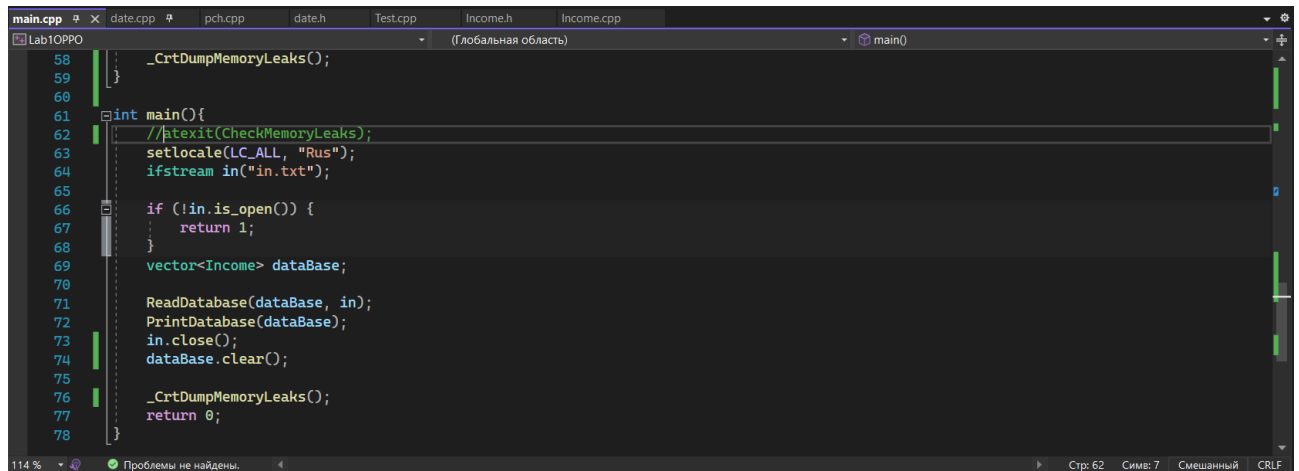


Рисунок 8 – Утечки обнаруженные CRT

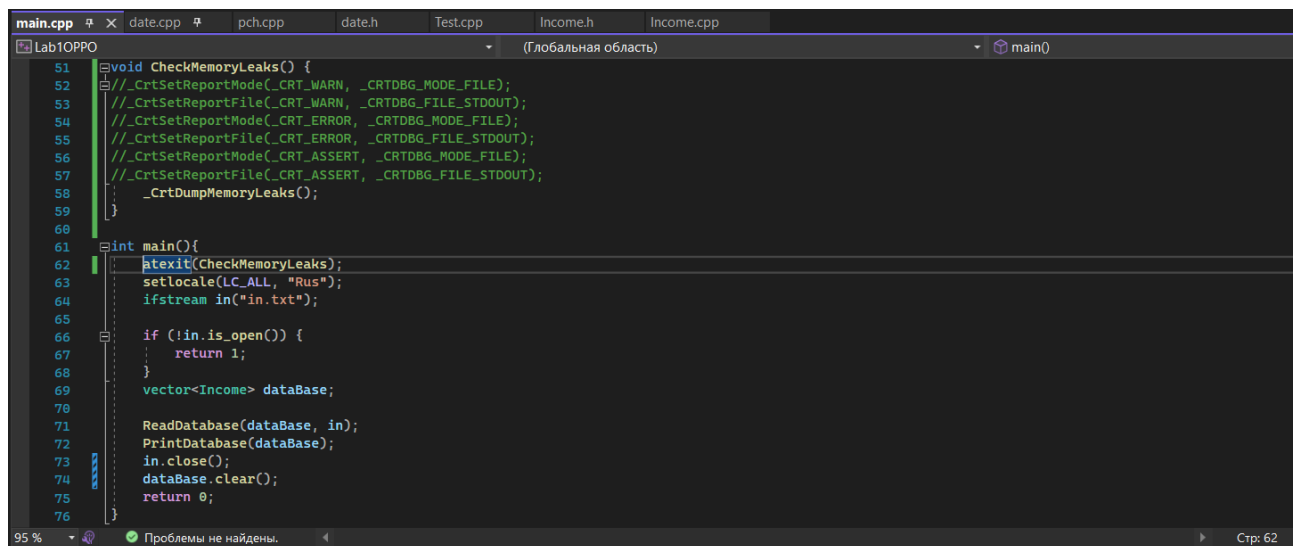
Утечки оказались не действительными. Они возникали из-за того, что функция `_CrtDumpMemoryLeaks()` вызывалась до выхода за пределы видимости `main` и `vector<Income> dataBase` не был очищен на момент её запуска. На Рисунке 9 код до исправления.



```
58     _CrtDumpMemoryLeaks();
59 }
60
61 int main(){
62     //atexit(CheckMemoryLeaks);
63     setlocale(LC_ALL, "Rus");
64     ifstream in("in.txt");
65
66     if (!in.is_open()) {
67         return 1;
68     }
69     vector<Income> dataBase;
70
71     ReadDatabase(dataBase, in);
72     PrintDatabase(dataBase);
73     in.close();
74     dataBase.clear();
75
76     _CrtDumpMemoryLeaks();
77     return 0;
78 }
```

Рисунок 9 – Код до исправления

На Рисунке 10 код после исправления.



```
51 void CheckMemoryLeaks() {
52     _CrtSetReportMode(_CRT_WARN, _CRTDBG_MODE_FILE);
53     _CrtSetReportFile(_CRT_WARN, _CRTDBG_FILE_STDOUT);
54     _CrtSetReportMode(_CRT_ERROR, _CRTDBG_MODE_FILE);
55     _CrtSetReportFile(_CRT_ERROR, _CRTDBG_FILE_STDOUT);
56     _CrtSetReportMode(_CRT_ASSERT, _CRTDBG_MODE_FILE);
57     _CrtSetReportFile(_CRT_ASSERT, _CRTDBG_FILE_STDOUT);
58     _CrtDumpMemoryLeaks();
59 }
60
61 int main(){
62     atexit(CheckMemoryLeaks);
63     setlocale(LC_ALL, "Rus");
64     ifstream in("in.txt");
65
66     if (!in.is_open()) {
67         return 1;
68     }
69     vector<Income> dataBase;
70
71     ReadDatabase(dataBase, in);
72     PrintDatabase(dataBase);
73     in.close();
74     dataBase.clear();
75     return 0;
76 }
```

Рисунок 10 – Код после исправления

После этих исправлений никаких ошибок не было обнаружено.