|  |
| --- |
| 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.

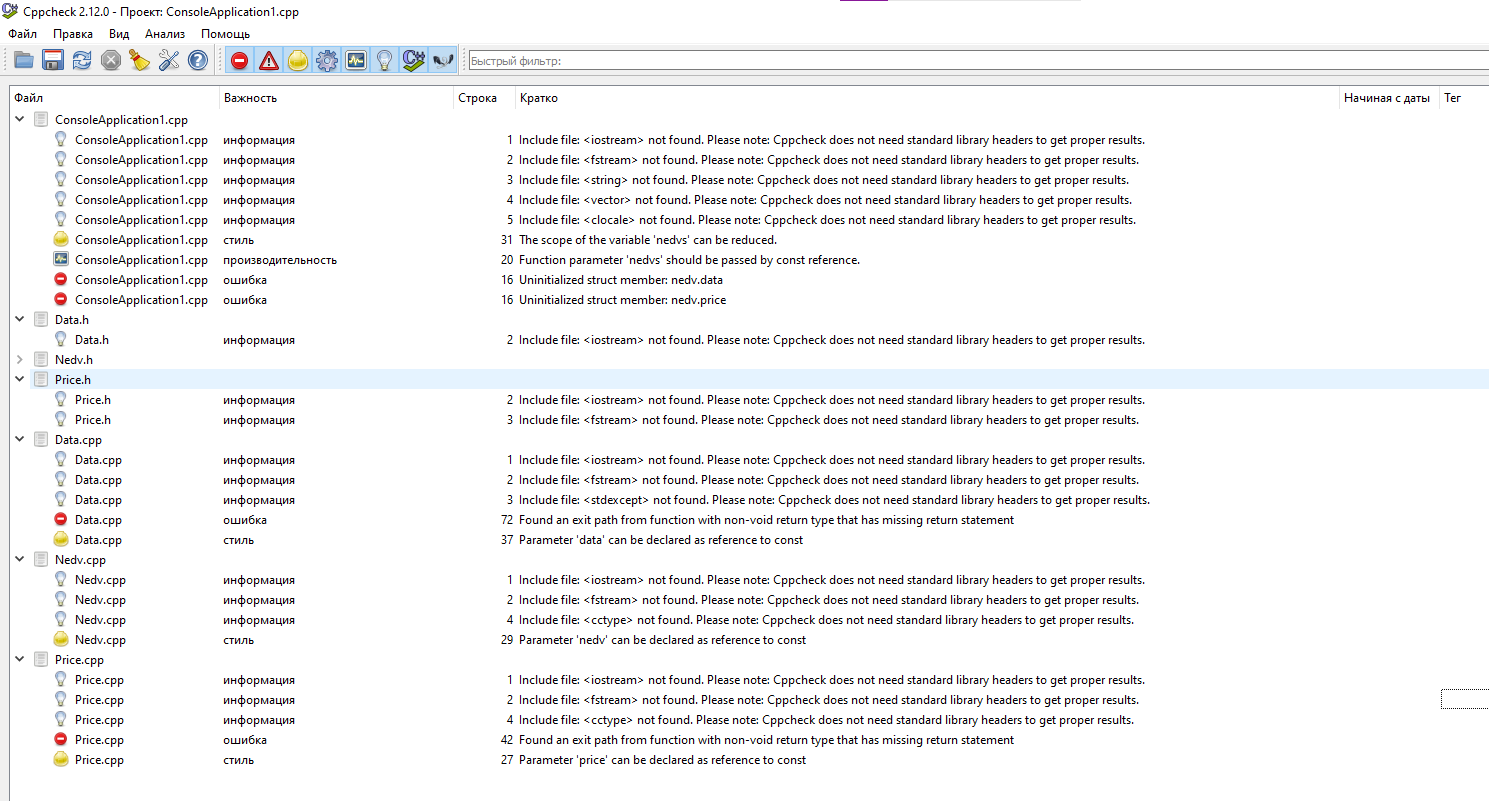
Таблица 1 – Задание по варианту

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

**Ход работы**

Пятую лабораторную работу я буду выполнять с помощью приложения «Cppcheck». Данное приложение показывает, имеющиеся ошибки, и описывает в чём именно проблема, а также где она находится.

Загрузив код, приложение показало, где имеются ошибки:



1. Анализ программы

**Примечания:**

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

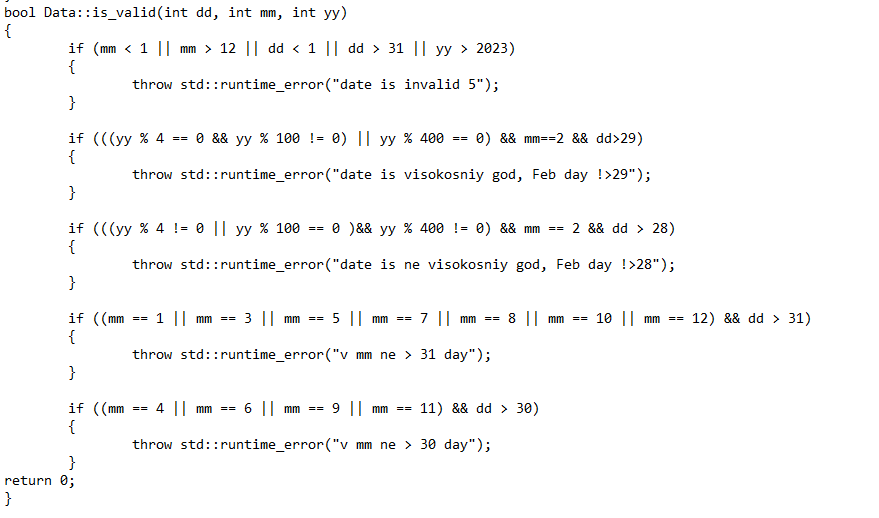
1. Информация – информационная важность показывает на то, что библиотеки, содержащиеся в файлах не нужны программе для чтения;
2. Ошибка – программа обнаружила ошибку в загруженных файлах, указала, что за ошибка обнаружена и предлагает свой вариант решения это проблемы;
3. Производительность – показывает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

* Вкладка «Строка»: показывает в какой именно строке была обнаружена ошибка;
* Вкладка «Кратко»: содержит общую информацию об ошибочности конкретной строки в конкретном файле;

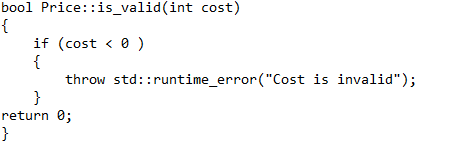
**Устранение полученных недочетов**

*Устранение ошибок «ошибка»:*

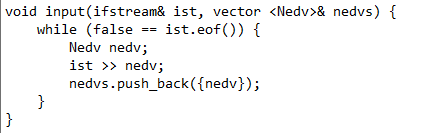
1. В файле «Data.cpp» в 72 строке имеется «ошибка», которая говорит нам о том, что отсутствует оператор возврата. Для устранения данной ошибки достаточно написать «return 0;» для пути выхода из функции с типом возврата non-void:



1. Устранение ошибки
2. В файле «Price.cpp» в 42 строке имеется «ошибка», которая говорит нам о том, что отсутствует оператор возврата. Для устранения данной ошибки достаточно написать «return 0;» для пути выхода из функции с типом возврата non-void:

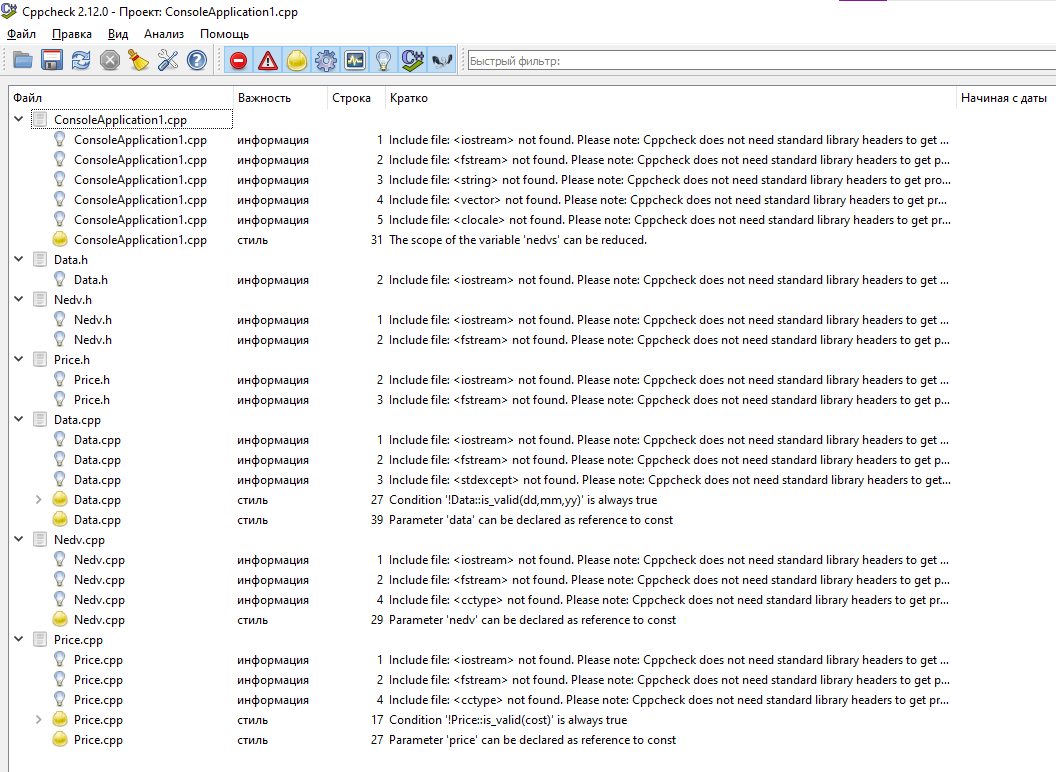


1. Устранение ошибки
2. В файле «ConsoleApplication1.cpp» имеются две одинаковые «ошибки». Чтобы избавиться от них нам необходимо «nedv» взять в фигурные скобки:



1. Устранение ошибки

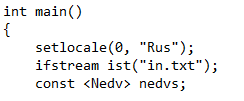
После устранения всех «ошибок» у нас остались только ошибки «стиль»:



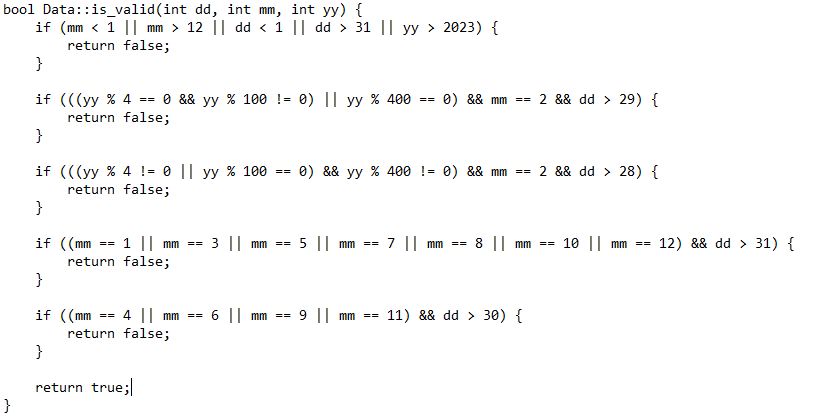
1. Устранение ошибки

*Устранение ошибок «стиль»:*

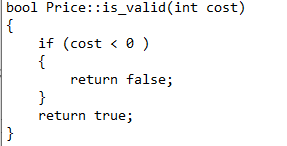
1. Чтобы устранить «стиль» в файле «ConsoleApplication1.cpp» нужно предать «nedvs» не «vector», а как «const»:



1. Устранение ошибки
2. В файле «Data.cpp», чтобы убрать ошибку «стиль» в строке 27, необходимо после каждого условия прописать «return false», а в конце прописать «return true»:

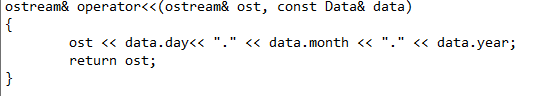


1. Устранение ошибки
2. В файле «Price.cpp», чтобы убрать ошибку «стиль» в строке 17, необходимо после условия прописать «return false», а в конце прописать «return true»:



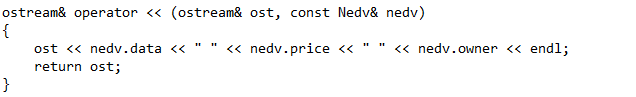
1. Устранение ошибки
2. В остальных «стилях» файлов: «Data.cpp», «Nedv.cpp» и «Price.cpp» обнаружена одна и таже ошибка. Соответствующие параметры должны быть объявлены как ссылки на «const»:

«Data.cpp»:



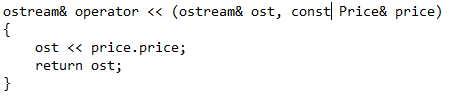
1. Устранение ошибки

«Nedv.cpp»:

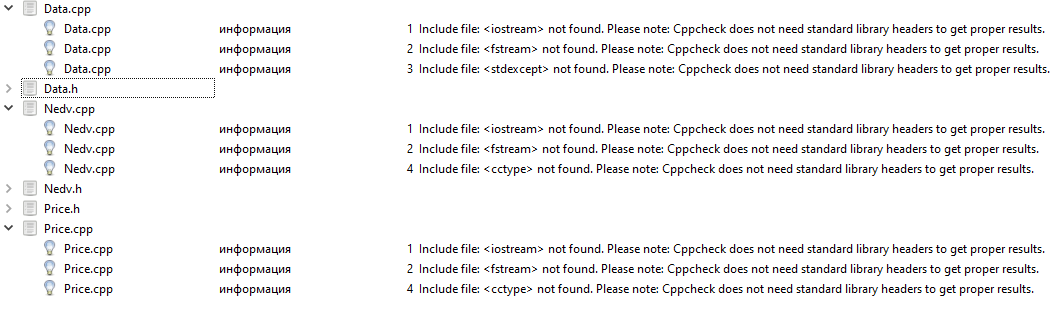


1. Устранение ошибки

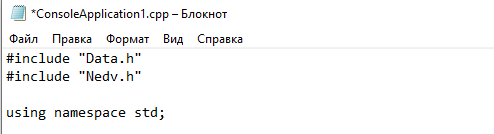
«Price.cpp»:



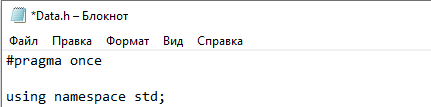
1. Устранение ошибки

После устранения всех ошибок важности «стиль», запускаем анализ программы: 

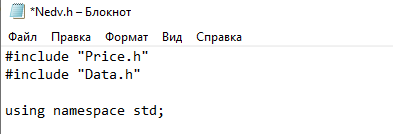
1. Запуск анализа программы
2. Во вкладке важность – критерий информация во всех файлах содержит одну и ту же проблему. Cppcheck не нуждается в заголовках стандартных библиотек для получения правильного результата. Для решения этих проблем, в каждом файле нужно удалить библиотеки.



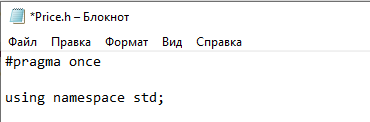
1. Устранение ошибки



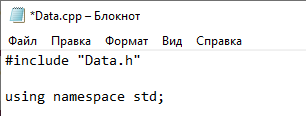
1. Устранение ошибки



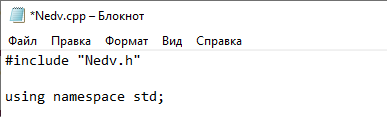
1. Устранение ошибки



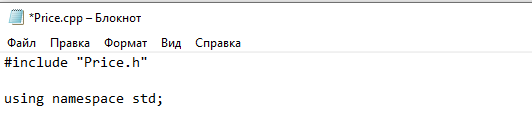
1. Устранение ошибки



1. Устранение ошибки

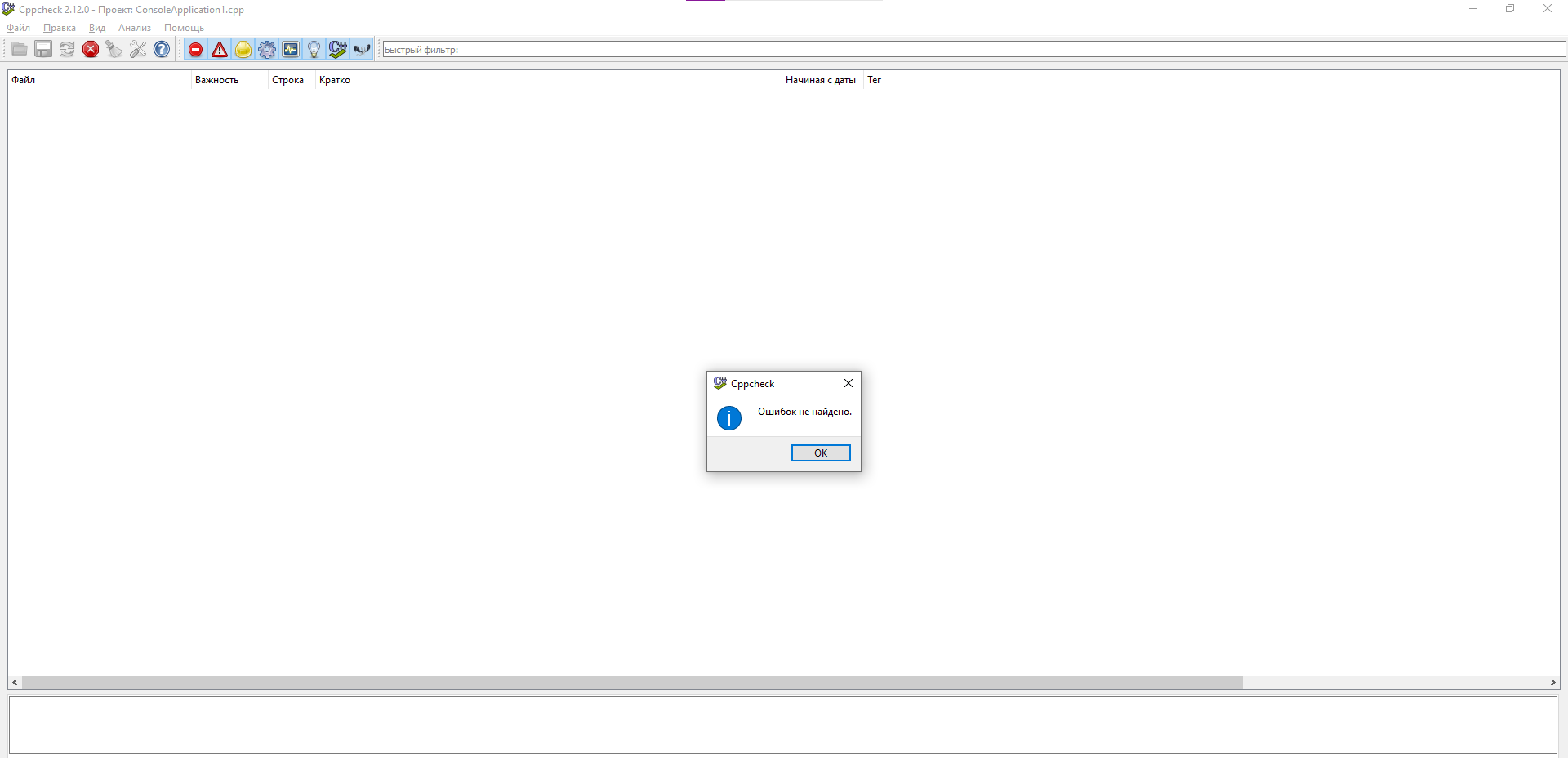


1. Устранение ошибки



1. Устранение ошибки

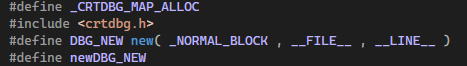
После устранения проблем различных важностей запускаем анализ программы:



1. Запуск анализа программы

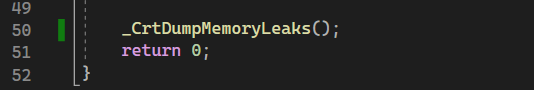
**Устранение полученных недочетов**

Для применения динамического анализатора я взяла пример утечки памяти. Для этого я использовала библиотеку CRT. Для её подключения в начале основного файла «ConsoleApplication1» прописываем данные команды:



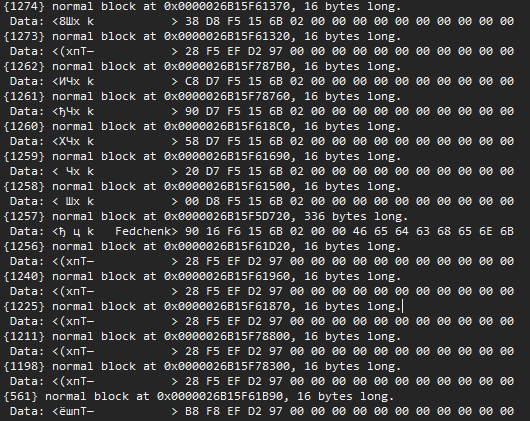
1. Применение CRT

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



1. Применение CRT

После запуска программы получим данные об утечки памяти:



1. Утечка памяти

**Заключение**

В процессе выполнения лабораторной работы я улучшила свои навыки отладки кода и исправления ошибок, а также приобрела новые знания по оценке качества программного обеспечения согласно установленным критериям. Моя цель была полностью достигнута - я создала программу, которая исправно работает и не содержит ошибок.