МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Кемеровский государственный университет»
Институт фундаментальных наук
Кафедра ЮНЕСКО по информационным вычислительным технологиям

ОТЧЕТ

по учебной практике, технологической (проектно-технологической) практике

проект «Инструменты анализа производительности программ C++»
(название проекта)

студентов 1 курса

Цариковой Дарьи Игоревны
(ФИО полностью)
Андрющенко Ивана Александровича
(ФИО полностью)

направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем.
направленность (профиль) подготовки «Информационные системы и базы данных».

Руководитель практики:
канд. физ.-мат. наук, доцент
К.С. Иванов
(ученая степень, звание, должность, ФИО)
Зав. кафедрой ЮНЕСКО по ИВТ
доктор физ.-мат. наук, профессор
Ю.Н. Захаров
(ученая степень, звание, должность, ФИО)

Работа защищена с оценками:

Царикова Д.И. (ФИО) — удовлетворительно (оценка) — «21» июня 2021 г.
Андрющенко И.А. (ФИО) — удовлетворительно (оценка) — «21» июня 2021 г.

Кемерово 2021

Ссылка на репозиторий: https://github.com/dariatsarikova/ProgecttesterC.git

Цель

Создать рабочее приложение используя ПО для анализа производительности программ на языке C++. Приложение: Калькулятор.

Задачи проекта

Проанализировать работу аналитического ПО, Изучить соответствующий теоретический материал, Закрепить навыки по работе с репозиторием Git.

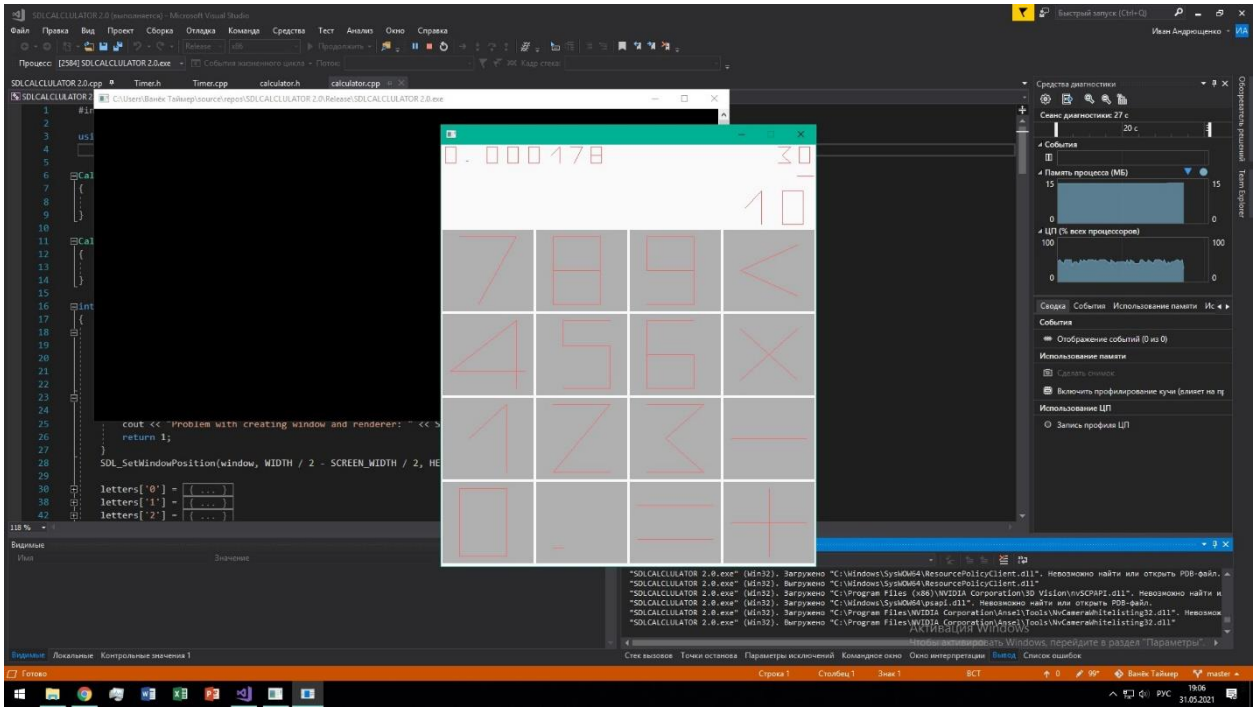
Актуальность проблемы

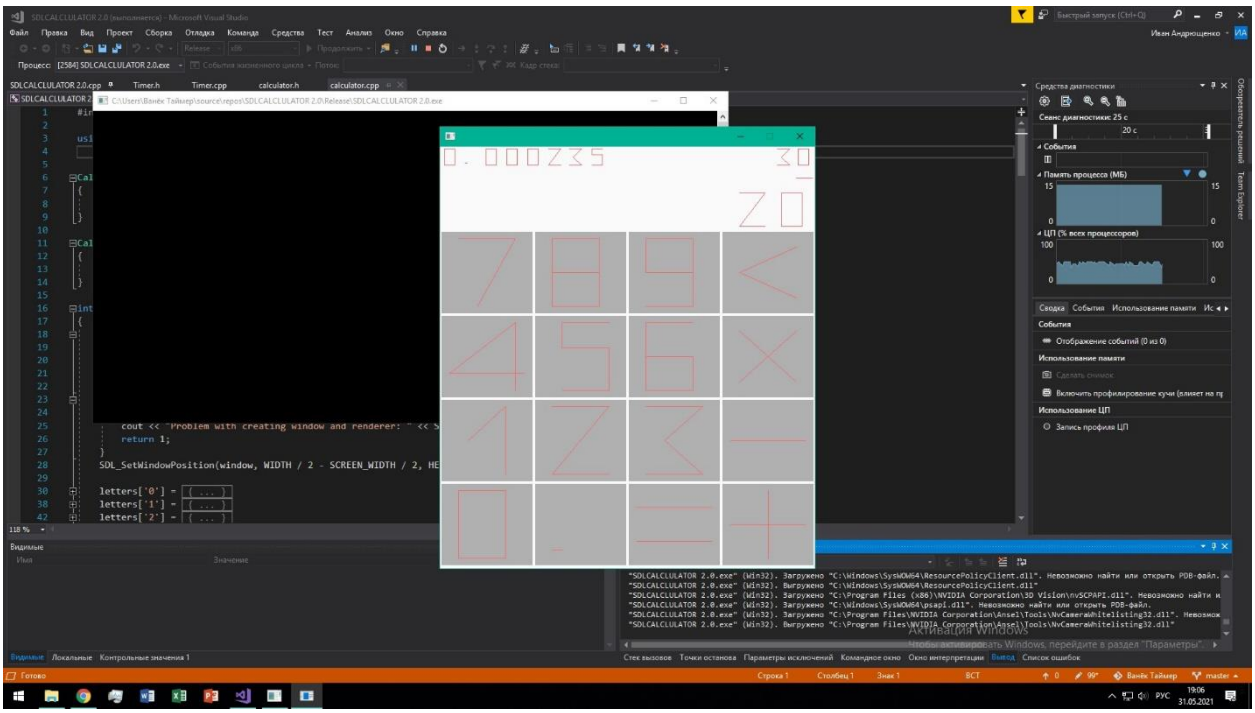
Данная проблема актуальна в рамках данной дисциплины уникальна. Позволит ознакомиться с аналитикой производительности, и с её использованием произвести продукт.

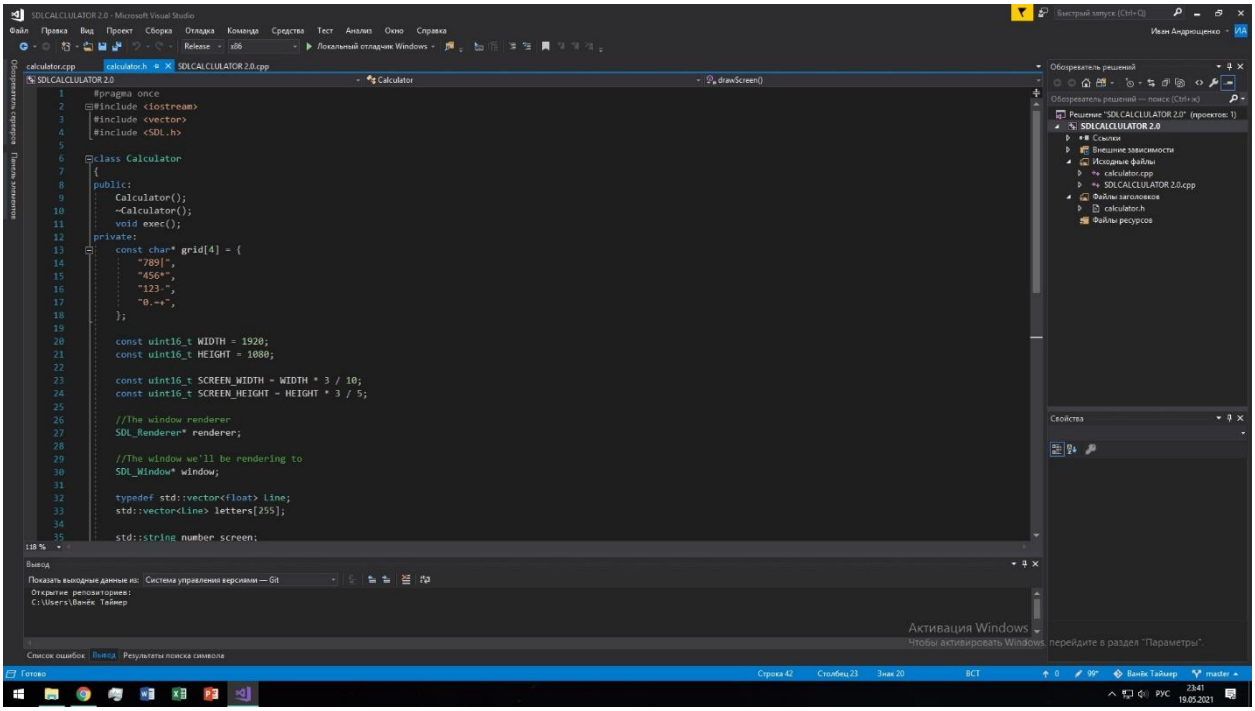
Ход работы:

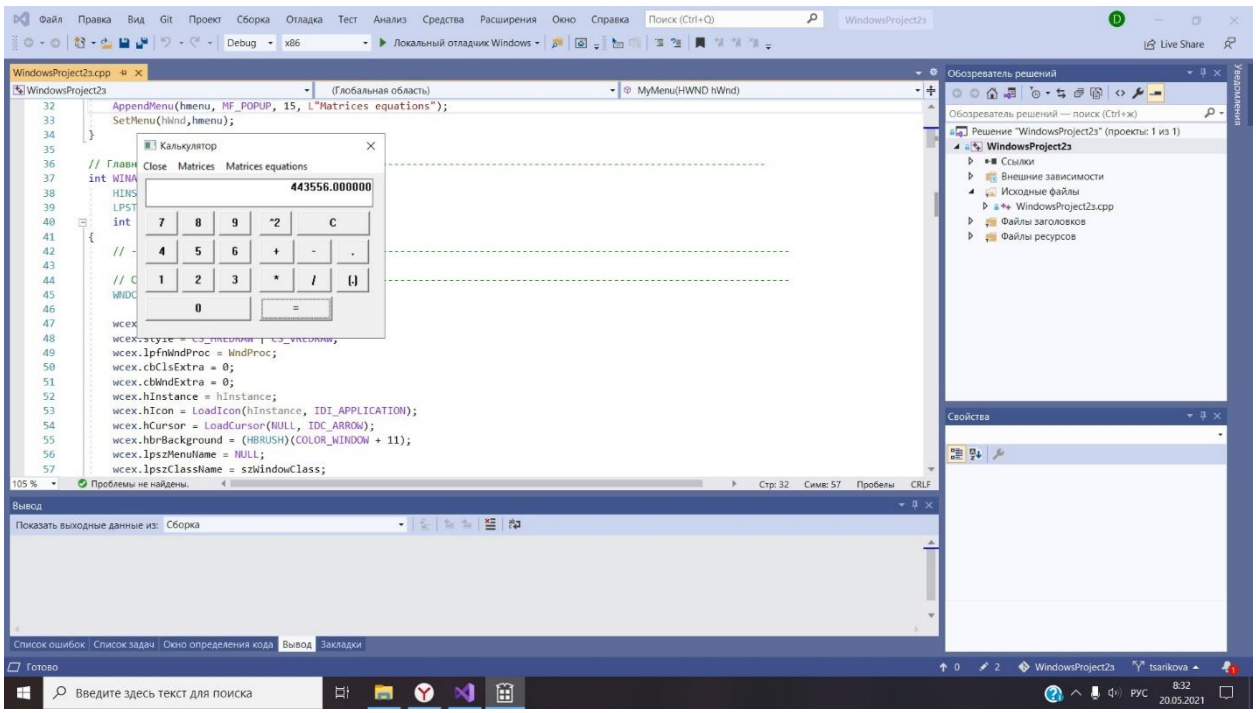
За время работы мы собрали основную теоретическую базу по использованию ПО анализа производительности, изучили её. Реализовали базовый функционал собственных приложений. Один из калькуляторов реализован на библиотеках SDL, а второй реализован с помощью WinAPI32. Основываясь на этом и на материале собственных лабораторных работ, провели следующее исследование.

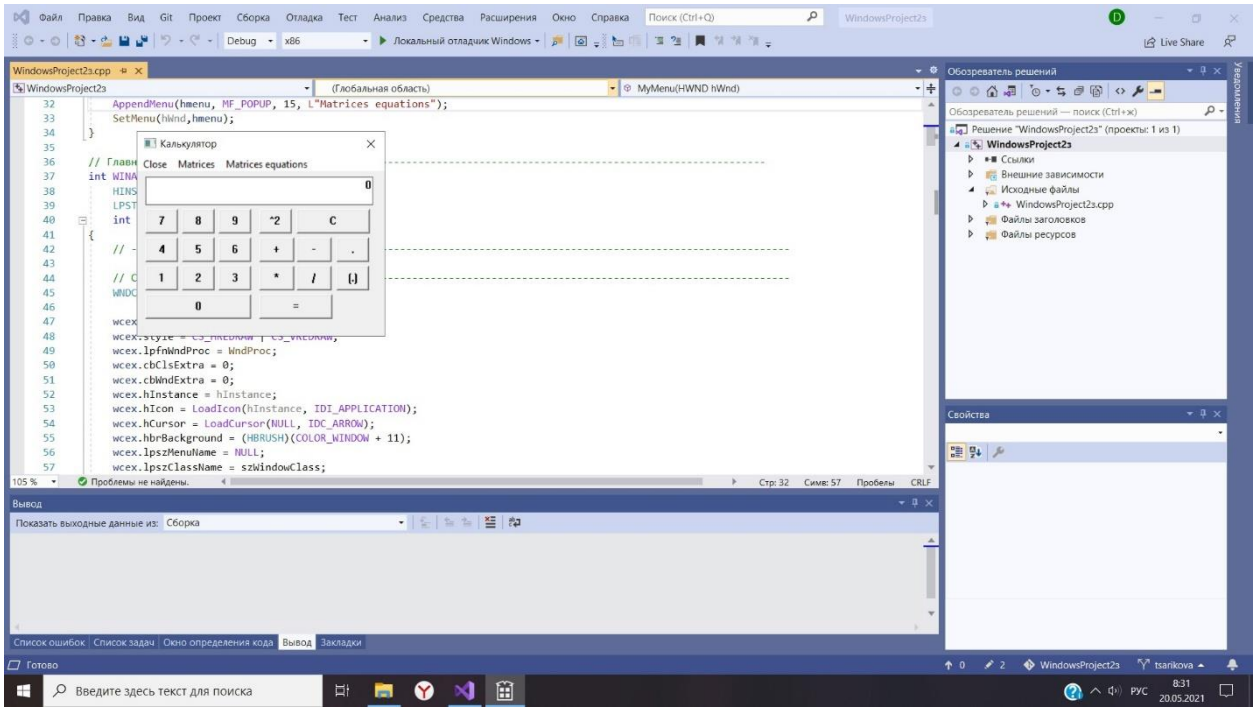
Демонстрация результатов практической деятельности:

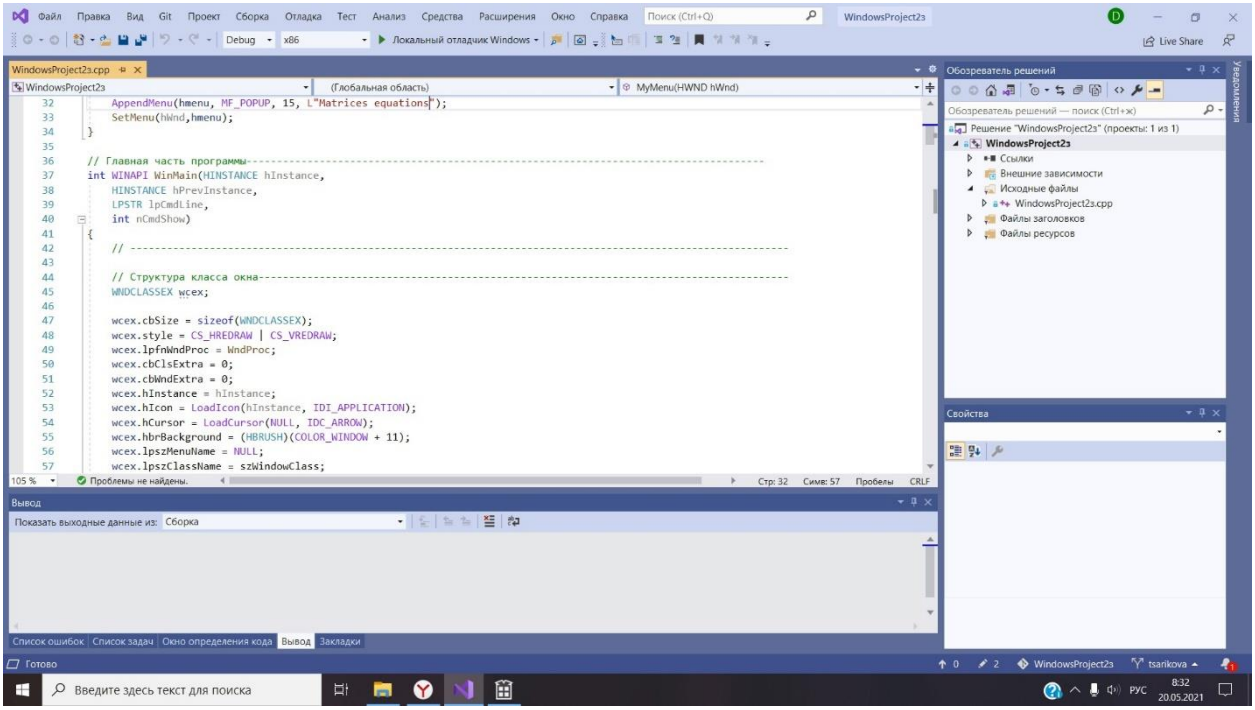







Файл Правка Вид Git Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
WindowsProject2s
Debug x86 Локальный отладчик Windows
Live Share
WindowsProject2s.cpp
(Глобальная область)
MyMenu(HWND hWnd)
AppendMenu(hmenu, MF_POPUP, 15, L"Matrices equations");
SetMenu(hWnd,hmenu);
Калькулятор
Close Matrices Matrices equations
443556.000000
7 8 9 *2 C
4 5 6 + - .
1 2 3 * / (-)
0 =
wcex.lpfnWndProc = WndProc;
wcex.cbClsExtra = 0;
wcex.cbWndExtra = 0;
wcex.hInstance = hInstance;
wcex.hIcon = LoadIcon(hInstance, IDI_APPLICATION);
wcex.hCursor = LoadCursor(NULL, IDC_ARROW);
wcex.hbrBackground = (HBRUSH)(COLOR_WINDOW + 11);
wcex.lpszMenuName = NULL;
wcex.lpszClassName = szWindowClass;
Обозреватель решений
Решение "WindowsProject2s" (проекты: 1 из 1)
WindowsProject2s
Ссылки
Внешние зависимости
Исходные файлы
WindowsProject2s.cpp
Файлы заголовков
Файлы ресурсов
Свойства
Вывод
Показать выходные данные из: Сборка
Список ошибок Список задач Окно определения кода Вывод Закладки
Готово
Введите здесь текст для поиска
8:32
20.05.2021


Файл Правка Вид Git Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Поиск (Ctrl+Q)
WindowsProject2s
Debug x86 Локальный отладчик Windows
Live Share
WindowsProject2s.cpp
(Глобальная область)
MyMenu(HWND hWnd)
AppendMenu(hmenu, MF_POPUP, 15, L"Matrices equations");
SetMenu(hWnd,hmenu);
Калькулятор
Close Matrices Matrices equations
0
7 8 9 *2 C
4 5 6 + - .
1 2 3 * / (-)
0 =
wcex.lpfnWndProc = WndProc;
wcex.cbClsExtra = 0;
wcex.cbWndExtra = 0;
wcex.hInstance = hInstance;
wcex.hIcon = LoadIcon(hInstance, IDI_APPLICATION);
wcex.hCursor = LoadCursor(NULL, IDC_ARROW);
wcex.hbrBackground = (HBRUSH)(COLOR_WINDOW + 11);
wcex.lpszMenuName = NULL;
wcex.lpszClassName = szWindowClass;
Обозреватель решений
Решение "WindowsProject2s" (проекты: 1 из 1)
WindowsProject2s
Ссылки
Внешние зависимости
Исходные файлы
WindowsProject2s.cpp
Файлы заголовков
Файлы ресурсов
Свойства
Вывод
Показать выходные данные из: Сборка
Список ошибок Список задач Окно определения кода Вывод Закладки
Готово
Введите здесь текст для поиска
8:31
20.05.2021



Программное обеспечение: Visual Studio 2019

Краткое руководство по запуску процессов

Запуск среды разработки Visual studio -> Открытие в ней проекта для анализа -> Выбрать в строке меню "Отладка" -> Выбрать "Профилировщик производительности" -> Выбрать необходимый критерий анализа -> Запустить тест.

Среда разработки Visual Studio позволяет анализировать производительность программного кода. Функция профилировщик производительности автоматически формирует отчёт.

Ниже представлен отчёт по приложению Калькулятор:

А) С точки зрения инструментирования

Критический путь

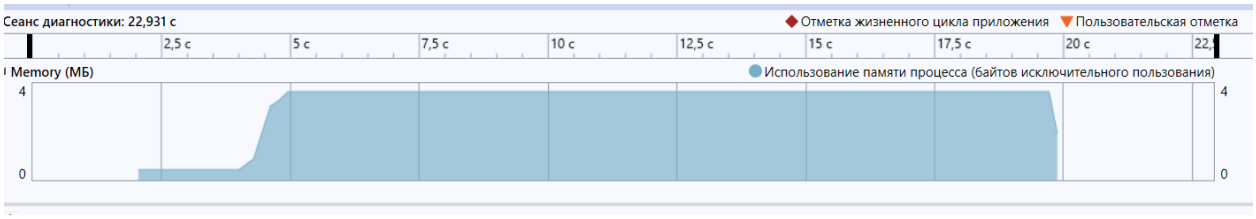
Имя функции	% затраченного инклюзивного времени	% затраченного эксклюзивного времени
__scrt_common_main	100.00	0.01
__scrt_common_main_seh	99.99	0.00
invoke_main	99.98	0.00
WinMain	99.98	0.00
GetMessageW	94.52	94.35

Связанные представления:Дерево вызовов Функции

Функции, выполняющие максимальную индивидуальную работу

Имя	Эксклюзивное время %
GetMessageW	94.35
NtdllDefWindowProc_W	2.97
CreateWindowExW	1.67
DispatchMessageW	0.42
UpdateWindow	0.35

Б) С точки зрения использования памяти



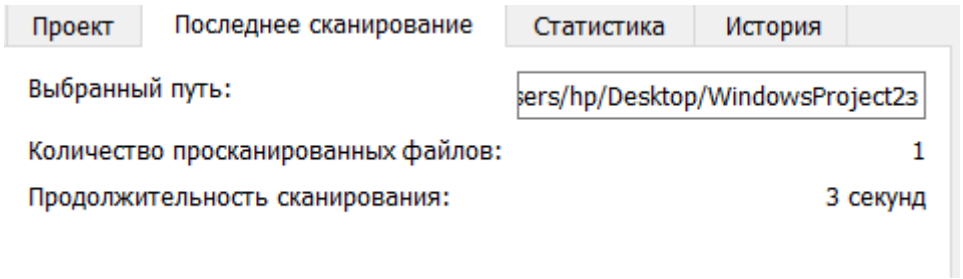
Ниже представлена проверка другого приложения, созданного в результате проектной деятельности

Программное обеспечение: Cppcheck 2.3

Краткое руководство по запуску процессов

Запуск CppCheck 2.3 -> Открытие в ней .cpp файла для анализа -> Выбрать в строке меню "Файл" -> Запустить тест

Ниже представлена статистика производительности проектного приложения



Ошибки: 0

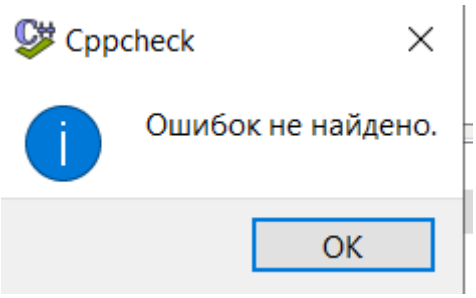
Предупреждения: 0

Стилистические предупреждения: 0

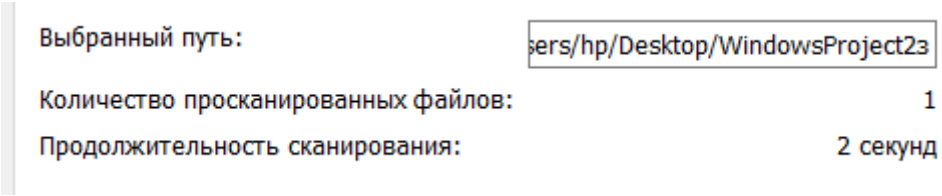
Предупреждения переносимости: 0

Проблемы с производительностью: 0

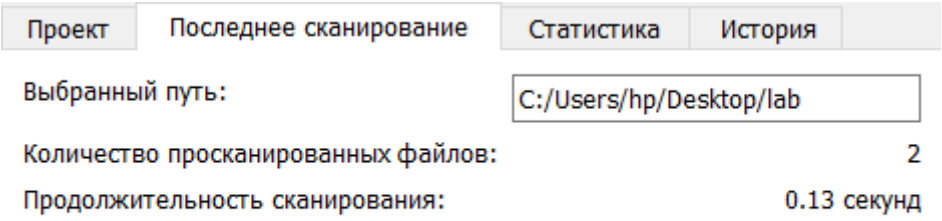
Информационные сообщения: 0

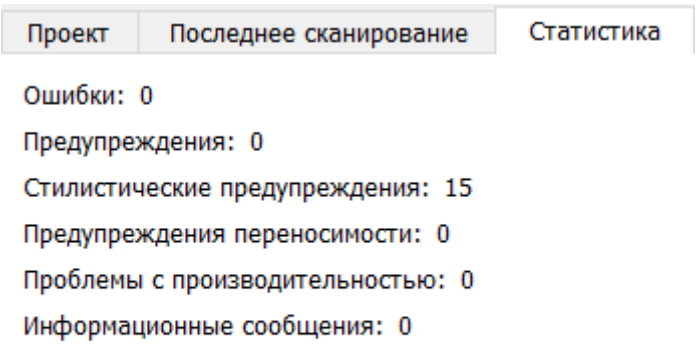


После выполнения даже незначительных изменений программного кода программа анализа работает корректно, выводит ошибки и недочёты. Но время анализа хоть и снижается, но всё же не так быстро как с консольными приложениями.

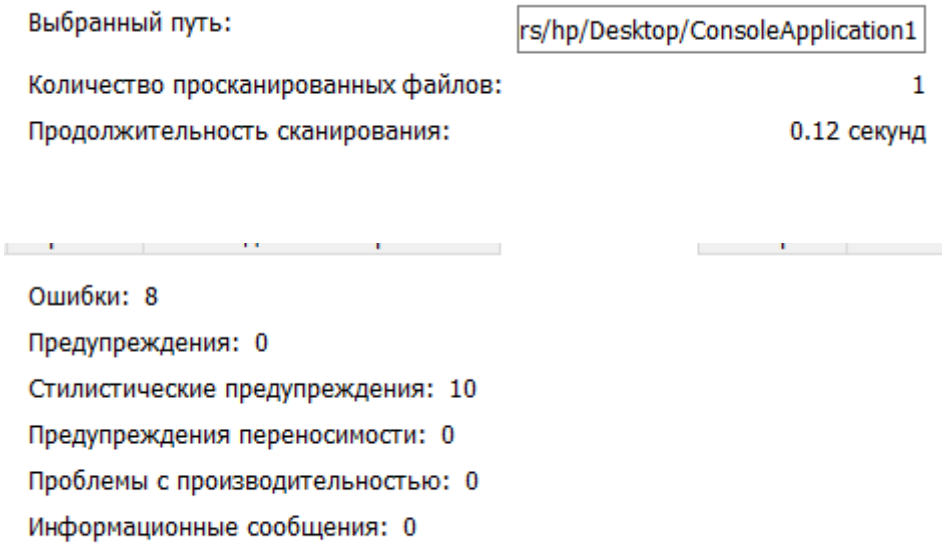


Ниже представлена проверка сданных лабораторных работ этого семестра. Приводятся из-за того, что на их основе строился код приложения.





Ниже приведена результаты неотлаженной функции решения Системы линейных алгебраических уравнений.



На основании этих данных соберём данные после поверхностного исследования в таблицу

Количество файлов	**Время обработки**	**Общее количество ошибок**	**Виды ошибок**	**Работает ли корректно**
1	3 секунды	0	-	Работает
2	0.13 секунд	15	Стилистические предупреждения	Работает
1	0.12 секунд	18	8-Ошибок 10-Стилистических предупреждений	Не работает

Сделаем выводы по работе самого ПО анализа производительности:

- Время обработки исходного кода зависит от: веса файла с кодом, количества файлов исходного кода, того какой раз запускается анализ одного и того же файла.
- Программный код может работать корректно, если отсутствуют грубые нарушения логики языка C++.
- Если сравнивать среду разработки Visual studio 2019 и ПО анализа производительности CppCheck 2.3, то первая более низкого уровня доступа к программного кода. К тому же используется запуск кода и тест в процессе работы приложения. Данная функция позволяет произвести более точный анализ.
- Также из плюсов среды разработки является формирование наглядного отчёта.

Список используемой литературы:

- **http://cppcheck.sourceforge.net/**
- **https://eax.me/c-static-analysis/**

- **https://scan.coverity.com/**
- **https://docs.microsoft.com/ru-ru/cpp/windows/walkthrough-creating-windows-desktop-applications-cpp?view=msvc-160&viewFallbackFrom=vs-2019**
- **https://docs.microsoft.com/ru-ru/visualstudio/profiling/profiling-feature-tour?view=vs-2019**