**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

Царикова Дарья Игоревна

(ФИО полностью)

Андрющенко Иван Александрович

(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:

« » 2021 г.

(ФИО) (оценка)

« » 2021 г.

(ФИО) (оценка)

Кемерово 2021

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

### Цель

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

### Задачи проекта

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

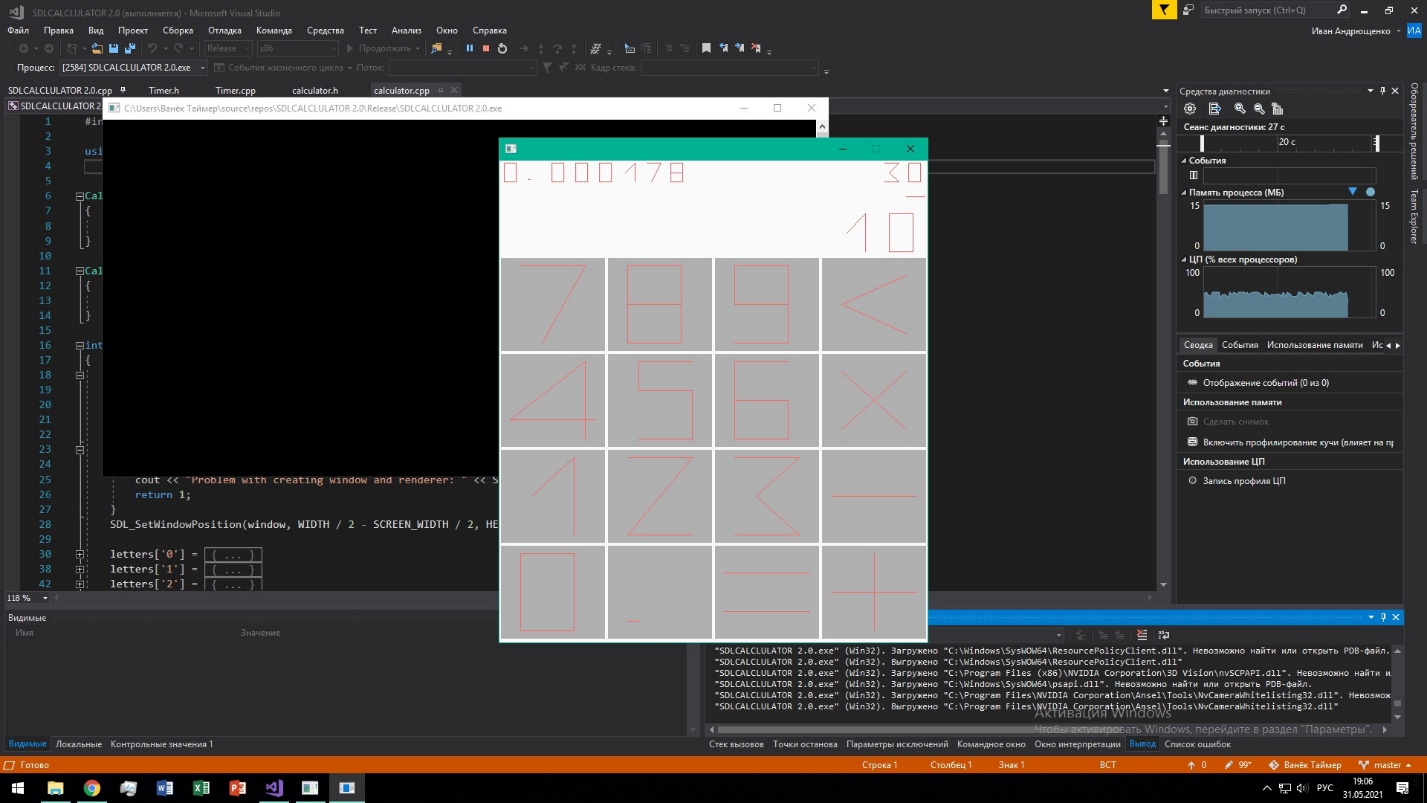
### Актуальность проблемы

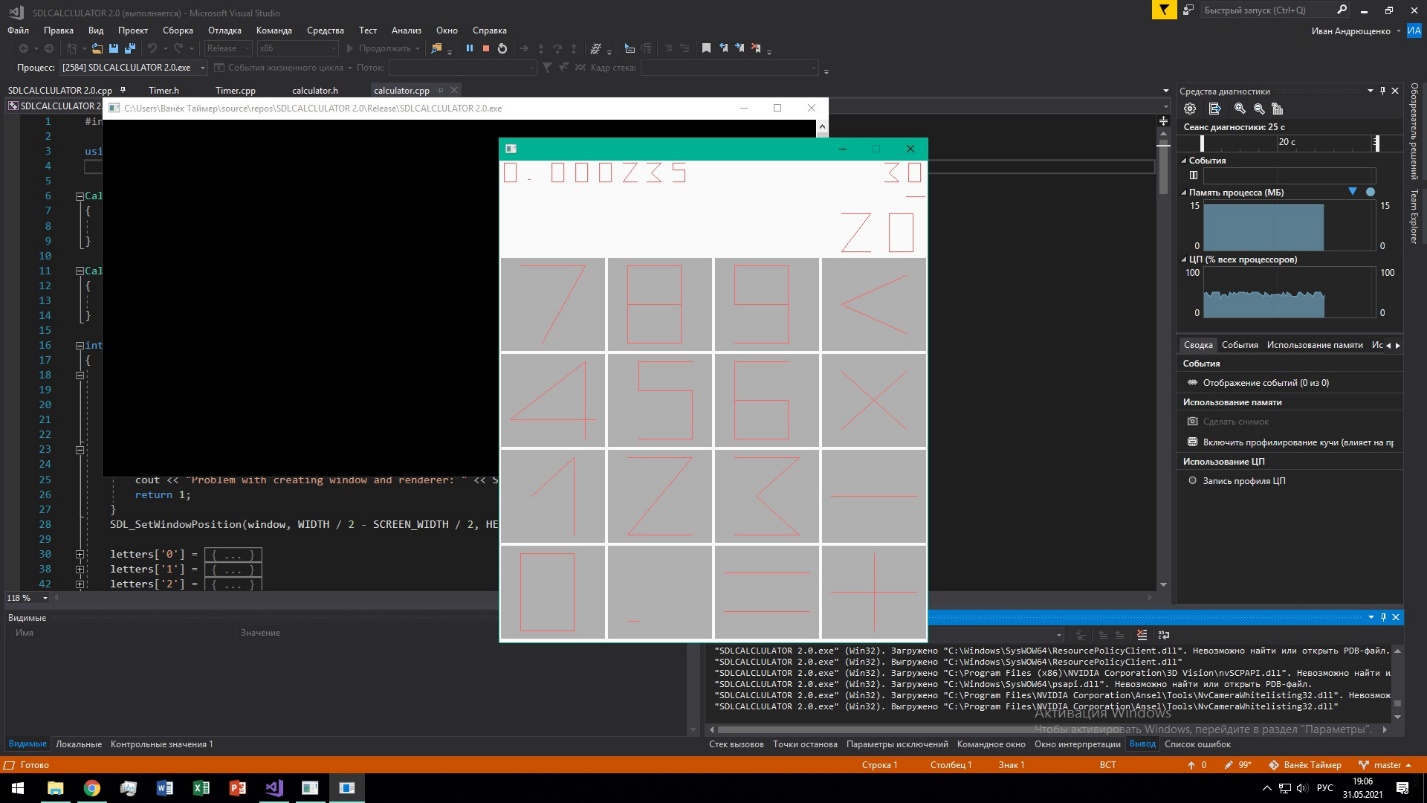
Данная проблема актуальна в рамках данной дисциплины уникальна. Позволит ознакомиться с аналитикой производительности, и с её использованием произвести продукт.

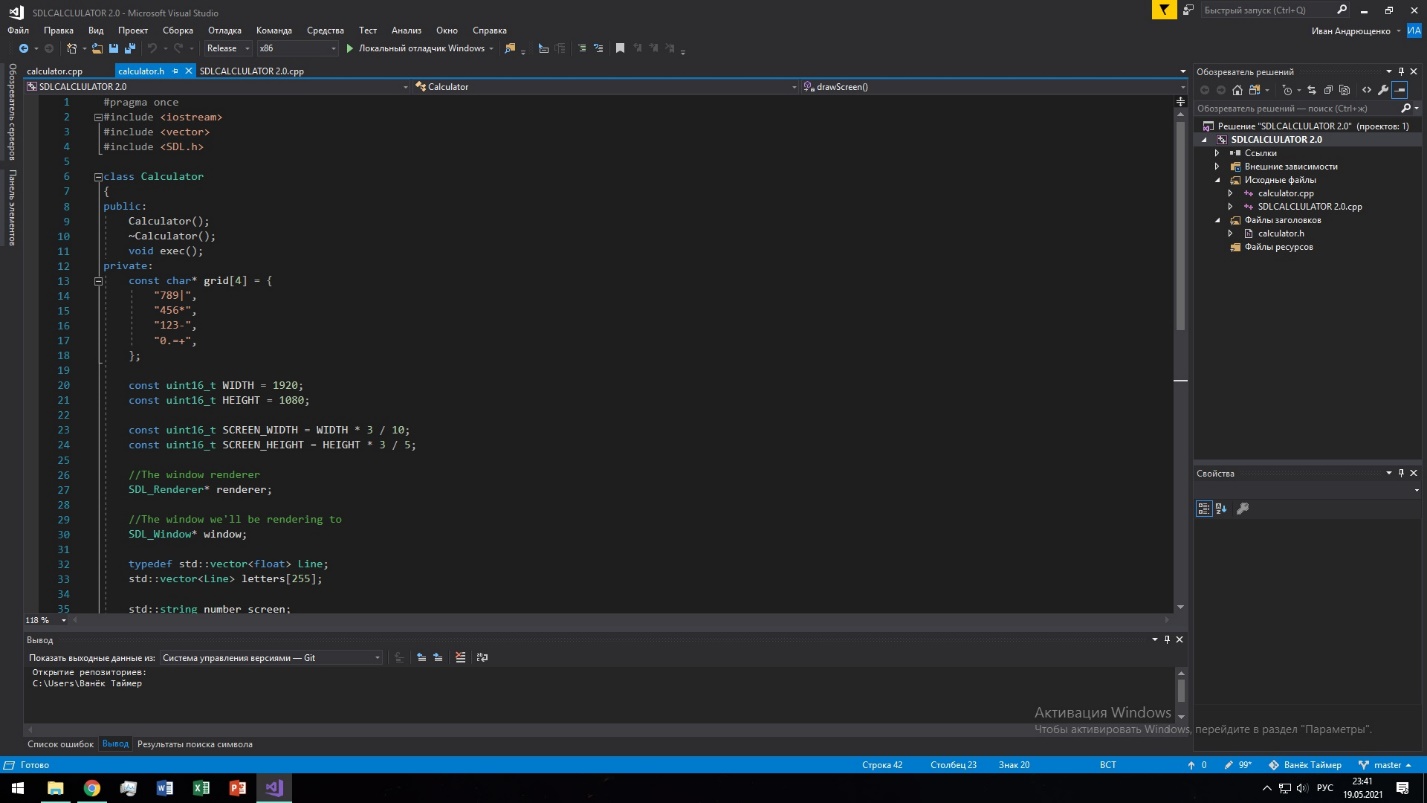
**Ход работы:**

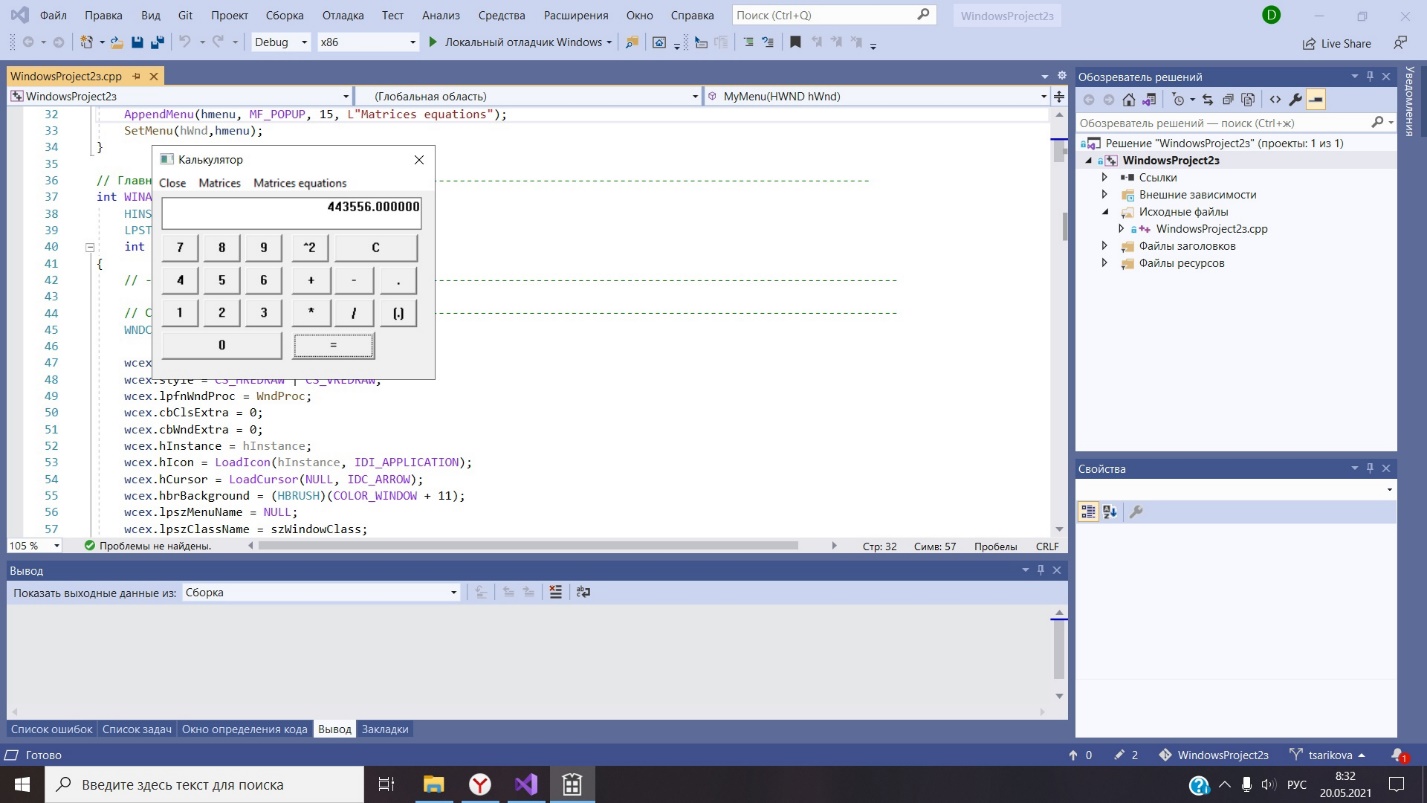
За время работы мы собрали основную теоретическую базу по использованию ПО анализа производительности, изучили её. Реализовали базовый функционал собственных приложений. Один из калькуляторов реализован на библиотеках SDL, а второй реализован с помощью WinAPI32. Основываясь на этом и на материале собственных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

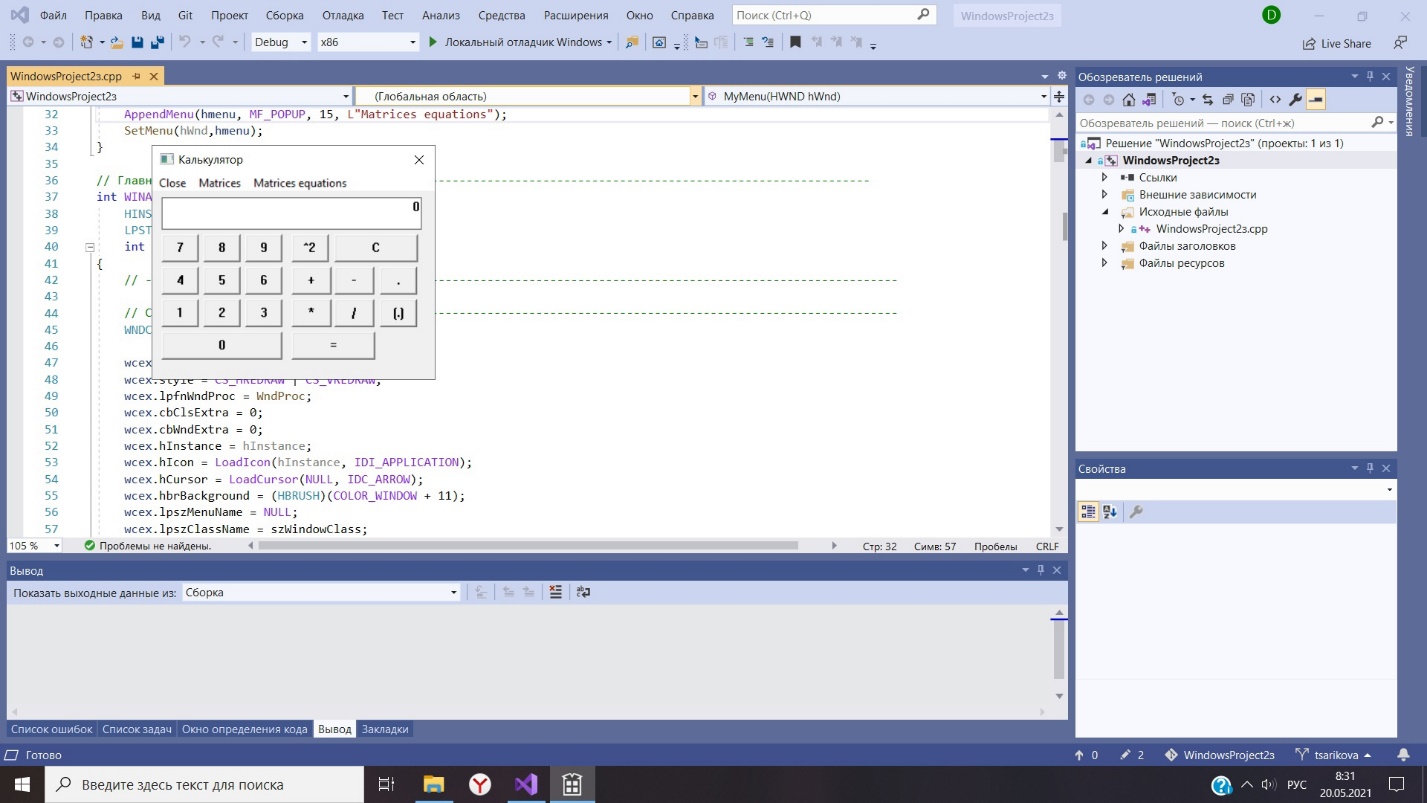
**Демонстрация результатов практической деятельности:**

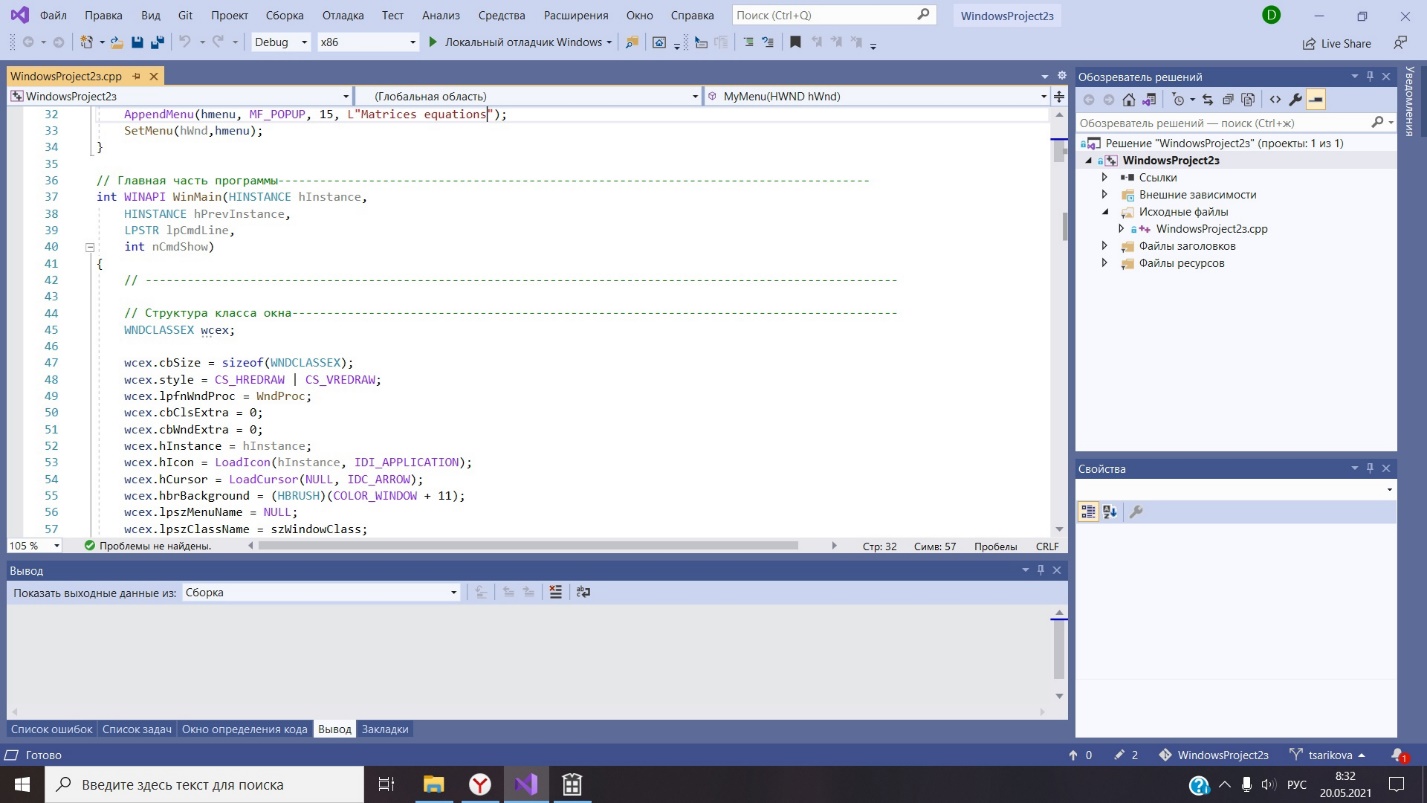












**Программное обеспечение: Visual Studio 2019**

***Краткое руководство по запуску процессов***

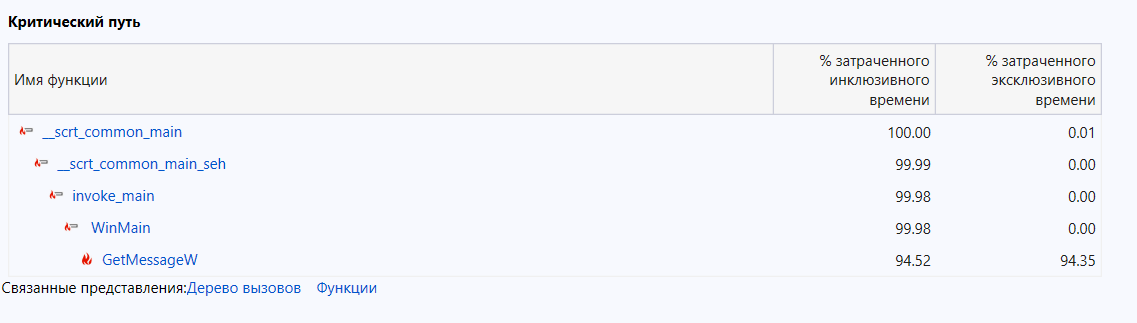
Запуск среды разработки Visual studio -> Открытие в ней проекта для анализа -> Выбрать в строке меню "Отладка" -> Выбрать "Профилировщик производительности" -> Выбрать необходимый критерий анализа -> Запустить тест.

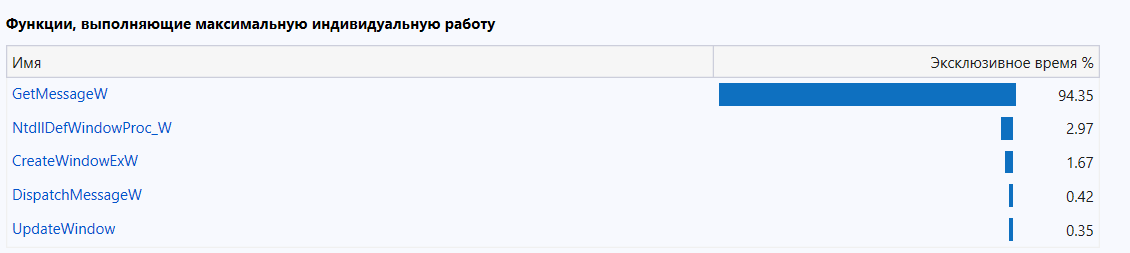
Среда разработки Visual Studio позволяет анализировать производительность программного кода. Функция профилировщик производительности автоматически формирует отчёт.

Ниже представлен отчёт по приложению Калькулятор:

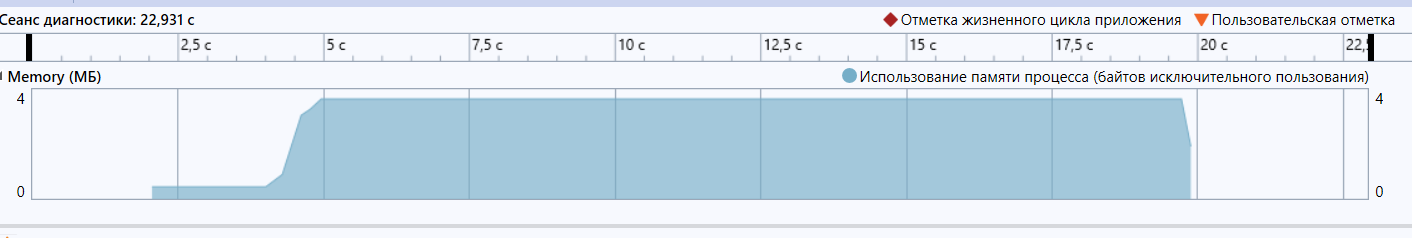
А) С точки зрения инструментирования







Б) С точки зрения использования памяти



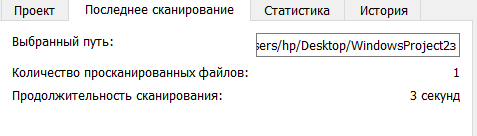
Ниже представлена проверка другого приложения, созданного в результате проектной деятельности

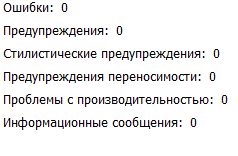
**Программное обеспечение: Cppcheck 2.3**

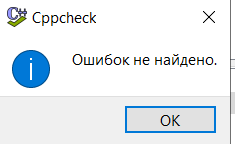
***Краткое руководство по запуску процессов***

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

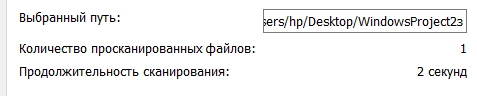
Ниже представлена статистика производительности проектного приложения



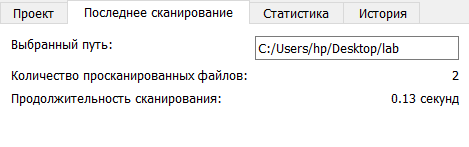


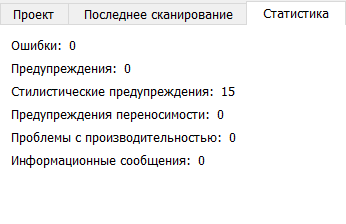


После выполнения даже незначительных изменений программного кода программа анализа работает корректно, выводит ошибки и недочёты. Но время анализа хоть и снижается, но всё же не так быстро как с консольными приложениями.

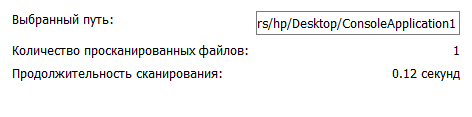


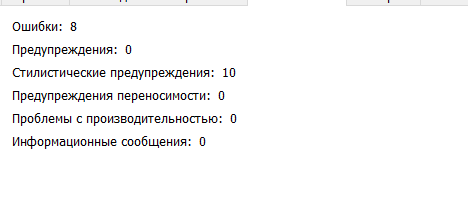
**Ниже представлена проверка сданных лабораторных работ этого семестра.** Приводятся из-за того, что на их основе строился код приложения.





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/**](http://cppcheck.sourceforge.net/)
* [**https://eax.me/c-static-analysis/**](https://eax.me/c-static-analysis/)
* [**https://scan.coverity.com/**](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/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**