МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХКОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра «Информатика»

Лабораторная работа №4

«Функции VС++ и консольные проекты

VisualStudio»

по дисциплине

«Разработка алгоритмов решения сложных задач методом

пошаговой детализации и их программная реализация»

Выполнил: студент гр. БЭИ2202 Тогузов А. А.

Вариант №26

Проверил: доц. Воробейчиков Л.А.

Москва, 2022 г.

Цель:

1. Изучить основные положения технологий структурного проектирования алгоритмов и метода пошаговой детализации, средства MS Visio для графической визуализации алгоритмов.
2. Выбрать вариант индивидуального задания из таблицы 4.1.
3. Проанализировать постановку задачи своего варианта индивидуального задания и, если необходимо, уточнить ее у преподавателя.
4. Провести формализацию решения задачи. Определить этапы ее решения с использованием метода пошаговой детализации.
5. Разработать схемы алгоритмов процедур на различных этапах метода пошаговой детализации и схему иерархии процедур.
6. Утвердить у преподавателя результаты выполнения п.п. 4-5.
7. Разработать программный код функций VC++ по алгоритмам п. 5
8. Создать консольный проект, содержащий три файла исходного кода: файл с главной функцией main, файл с функциями ввода и вывода данных и файл с функциями, решающими предписанные задачи. Обмен

данными между функциями должен осуществляться через параметры и возвращаемые значения, без использования глобальных переменных. Главная функция main должна содержать только операторы вызова разработанных функций.

1. Подготовить варианты исходных данных для тестирования проекта.
2. Выполнить проект с тестовыми исходными данными и получить результаты. Проверить правильность результатов.
3. Оформить отчет по работе в среде MS Word. Изобразить все схемы, используя средства MS Visio.
4. Представить преподавателю отчет по работе.
5. Ответить на замечания преподавателя по выполненной работе и на заданные им вопросы по теме.
6. Получить отметку о выполнении и защите работы.
7. Индивидуальное задание

Определите все углы треугольника при заданных значениях сторон a, b, c. Угол, лежащий против стороны a, вычисляется по теореме косинусов:

α = arccos(b2 +c2 – a2)/(2∙b∙c)

1. Формализация задания

Из условия задачи видно, что для вычисления треух углов понадобится три разных выражения.Таким образом, для вычисления формул на нижнем уровнеиерархии целесообразно использовать три общих процедур, вычисляющие углы треугольника.

1. Разработка алгоритмов решения задач
   1. На самом верхнем (первом) уровне алгоритм решения задачи можно укрупнено представить в виде вызова главной процедуры с именем main (Рисунок 1), где Ugly(процедура) вычисления значений углов a, b, c, Рисунок 1.

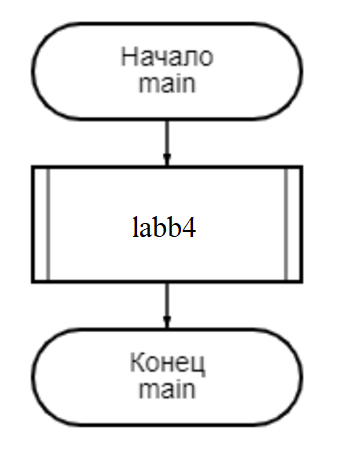


Рисунок 1 - Укрупненная схема алгоритма main решения задачи 26

* 1. На следующем, втором уровне, детализируем алгоритм процедуры Ugly путем представления его в виде последовательности следующих трех процедур, Рисунок 2
* процедуры ввода значений a, b, c с именем Getabc;
* процедуры вычисления A, B, C с именем VichisABC;
* процедуры вывода вычисленных значений углов, VivodABC

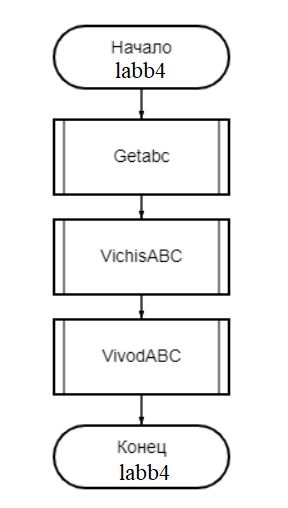


Рисунок 2 – Результат второго уровня детализации алгоритма

* 1. Перейдем к следующему, третьему уровню детализации. Процедуры Getabc и VivodABC дальнейшей детализации не требуют, так как средства ввода-вывода имеются в любом языке программирования. Поэтому на следующем, третьем шаге детализируем алгоритм процедуры вычисления A, B, C, VichisABC, Рисунок 3

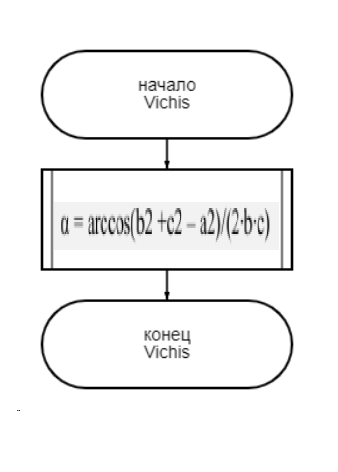


Рисунок 3 - Схема алгоритма процедуры VichisABC

Из расчетных формул видно, что для вычисления A, B, C необходимо вычислить значения углов заданной формулой. На этом уровне детализации воспользуемся формулой нахождение угола.

Схема иерархии процедур для решения задачи изображена на рисунке 7.

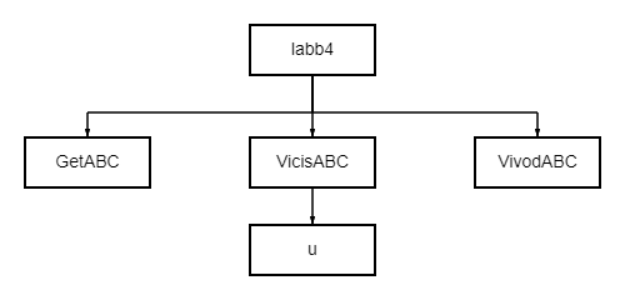


Рисунок 7 – Схема алгоритма

1. Разработка программного проекта, Рисунок 8 - 9

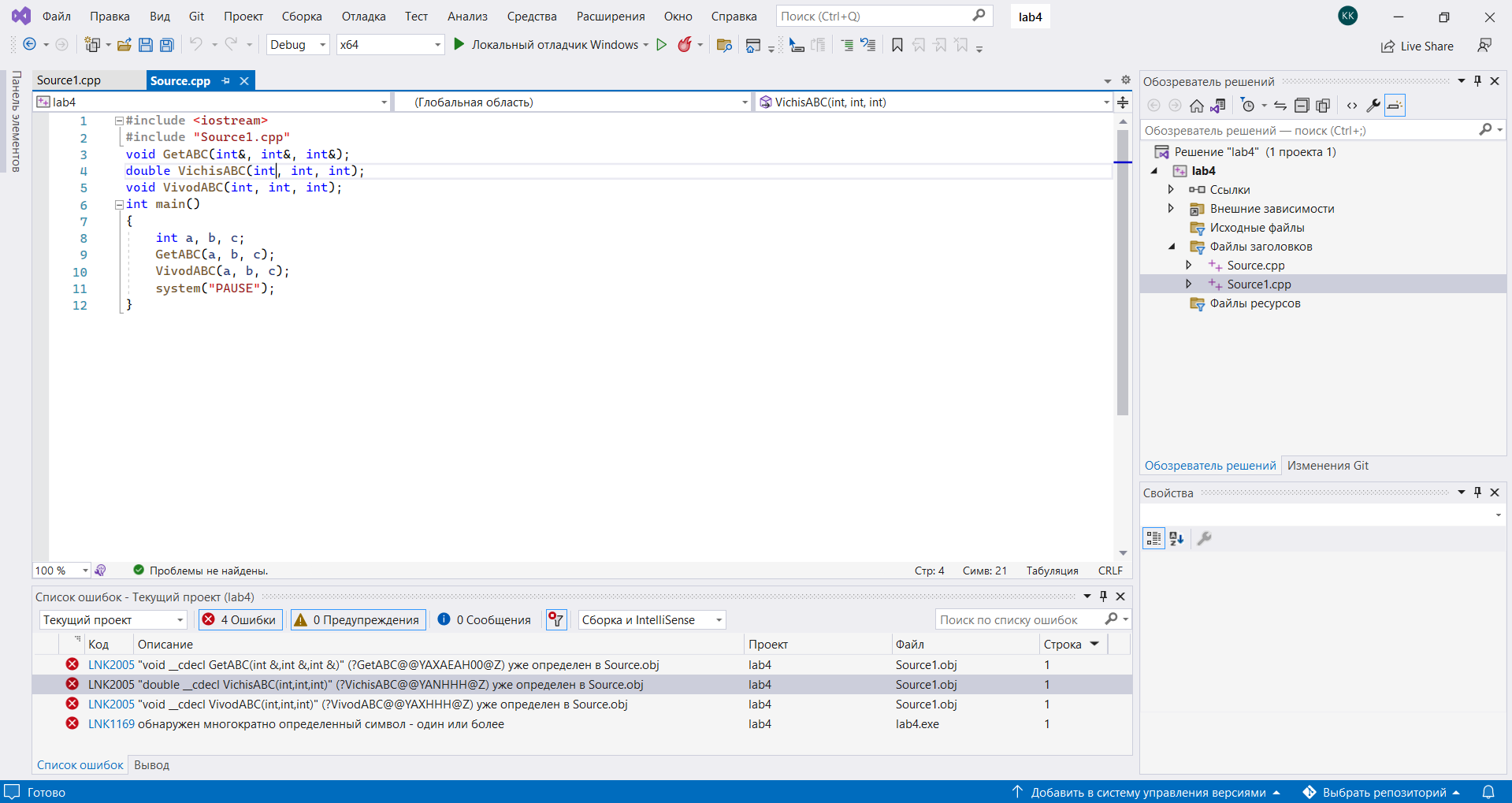


Рисунок 8 – программный код

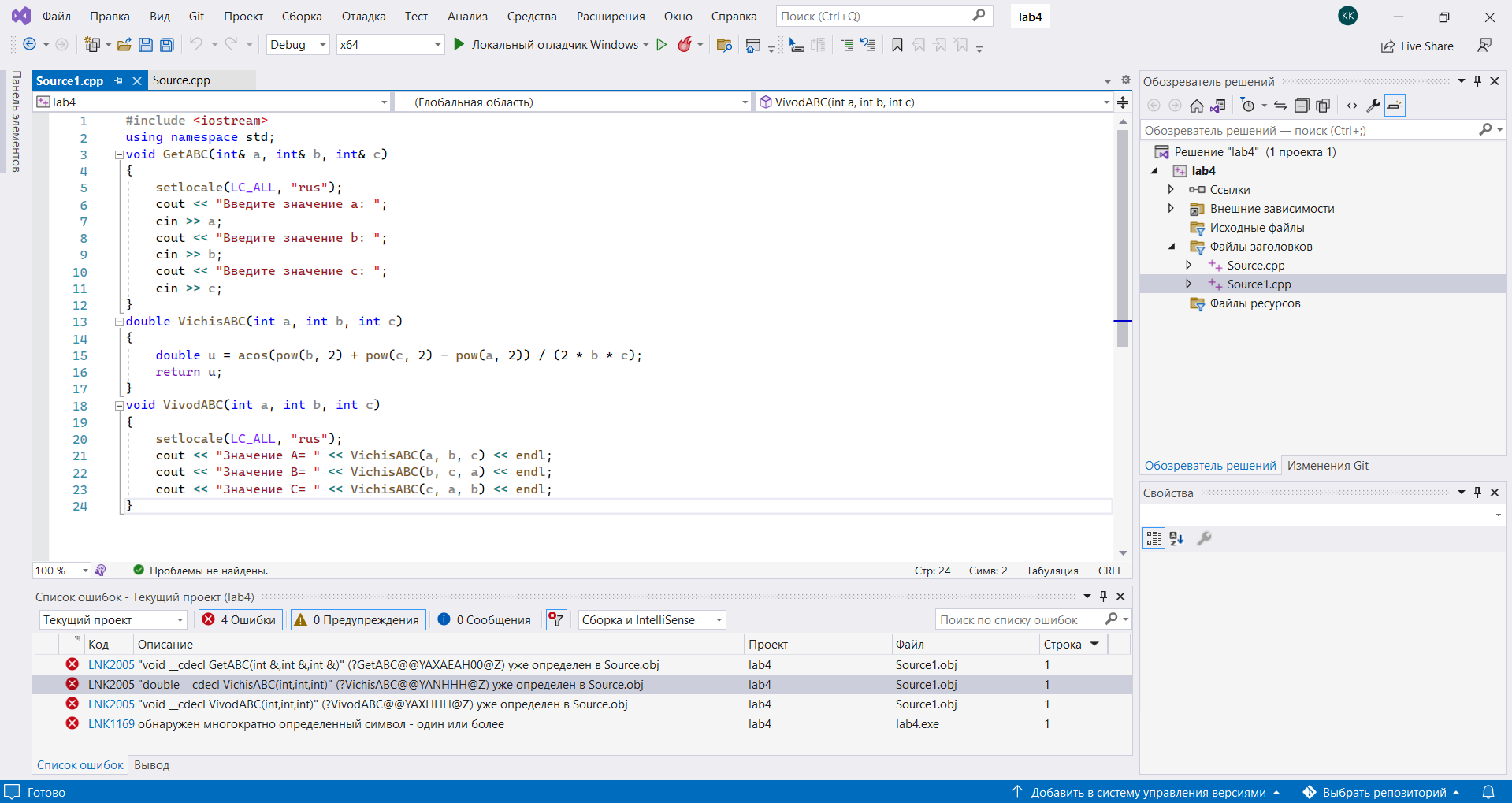


Рисунок 9 – программный код

1. Исходные данные для тестирования проекта.

Протестируем проект при двух значениях входных переменных a, b, c: a=3 и b=4 и с=5, a = 16 и b=18 и c = 20.

1. Результаты выполнения проекта, Рисунок 10 - 11

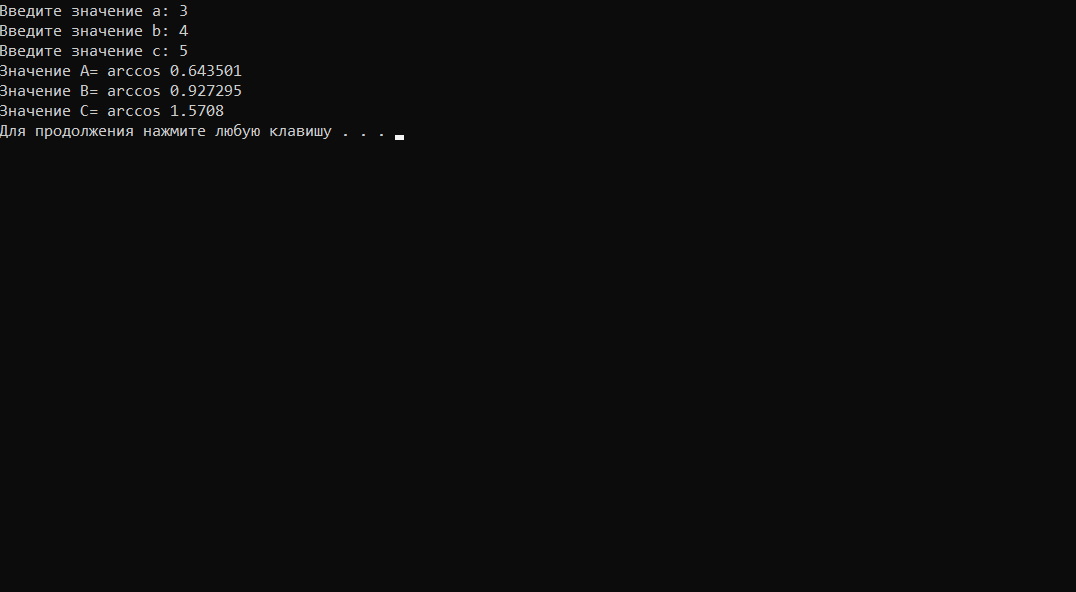


Рисунок 10 – полученные данные

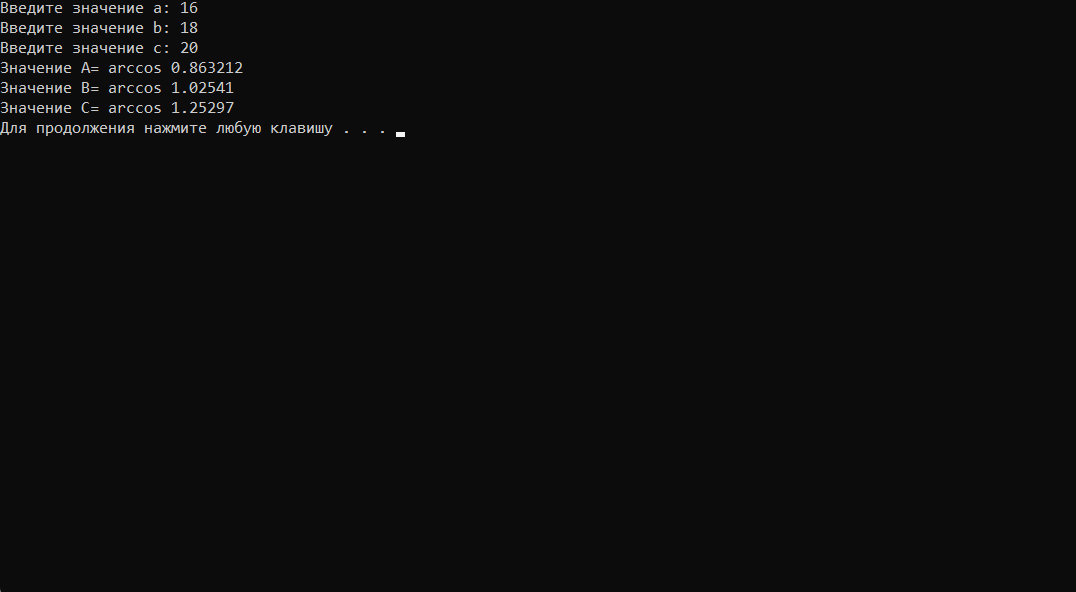


Рисунок 11 – полученные данные

1. Доказательство правильности результатов выполнения проекта

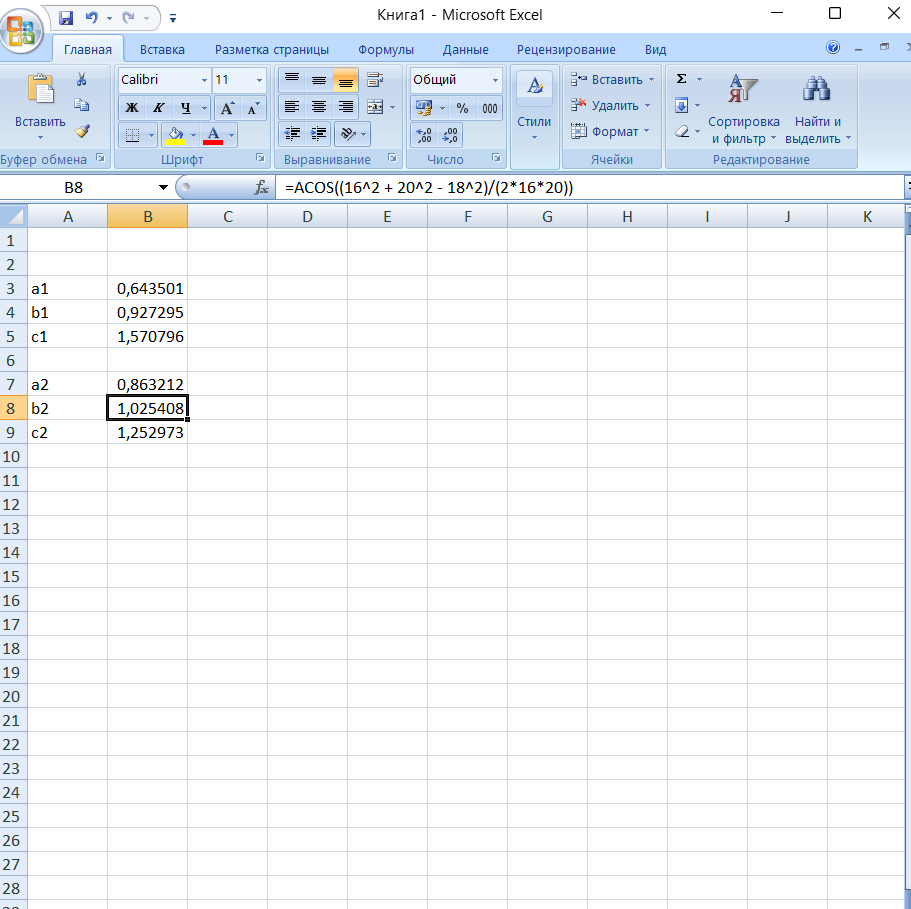


Рисунок 12 - Доказательство