Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное   
учреждение высшего образования

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе**

**«Вычисление математических функций с помощью рядов Тейлора»**

**Выполнила**:

студентка группы 3822Б1ПМ1

Жесткова А. Е.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2023

**Содержание**

[Постановка задачи](#_Toc26962562) 3

[Метод решения 4](#_Toc26962563)

[Руководство пользователя](#_Toc26962564) 5

[Описание программной реализации](#_Toc26962565) 6

[Подтверждение корректности](#_Toc26962566) 9

[Результаты экспериментов](#_Toc26962567) 10

[Заключение](#_Toc26962568) 11

[Приложение 1](#_Toc26962569)2

# Постановка задачи

Главной целью лабораторной работы является реализация подсчёта четырёх математический функций: sin(x), cos(x), ln(1+x) и ex.

Задачи лабораторной работы:

* Подсчёт математических функций при помощи рядов Тейлора прямым и обратным суммированием элементов;
* Проверка корректного счёта программы при помощи встроенных функций;
* Проведение эксперимента по определению точности вычисления функций;
* Формулировка вывода по всей работе программы.

# Метод решения

В программе нужно вычислить значение функций sin(x), cos(x), ln(1+x), ex. Подсчет будет основан с помощью разложения функций в ряд Тейлора, при этом слагаемые суммы будут вычисляться по рекуррентной формуле, которая для каждой функции будет задана по-разному.

Суммирование элементов рядов Тейлора будет осуществляться двумя способами: прямой и обратной суммой. Прямая сумма подразумевает суммирование элементов с первого, а обратная сумма - с последнего. Для обратной суммы также предусмотрены рекуррентные формулы, разные для всех функций.

# Руководство пользователя

После запуска программы пользователю предлагают ввести значение х – точку, в которой будет вычисляться значения математической функций.

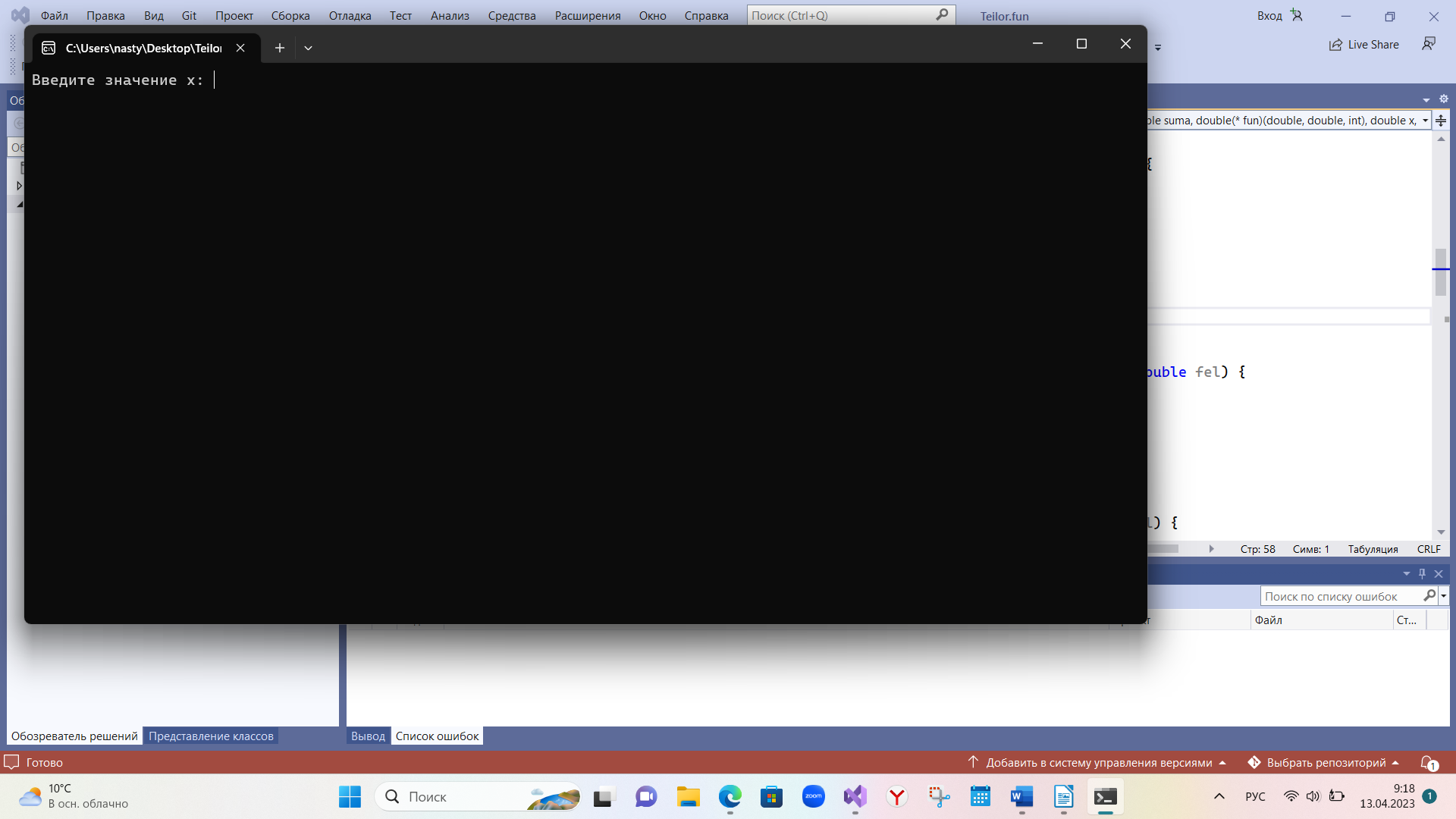


Рис. 1 «Главное меню 1»

Затем пользователь может выбрать одну из четырёх функций, которую программа посчитает.

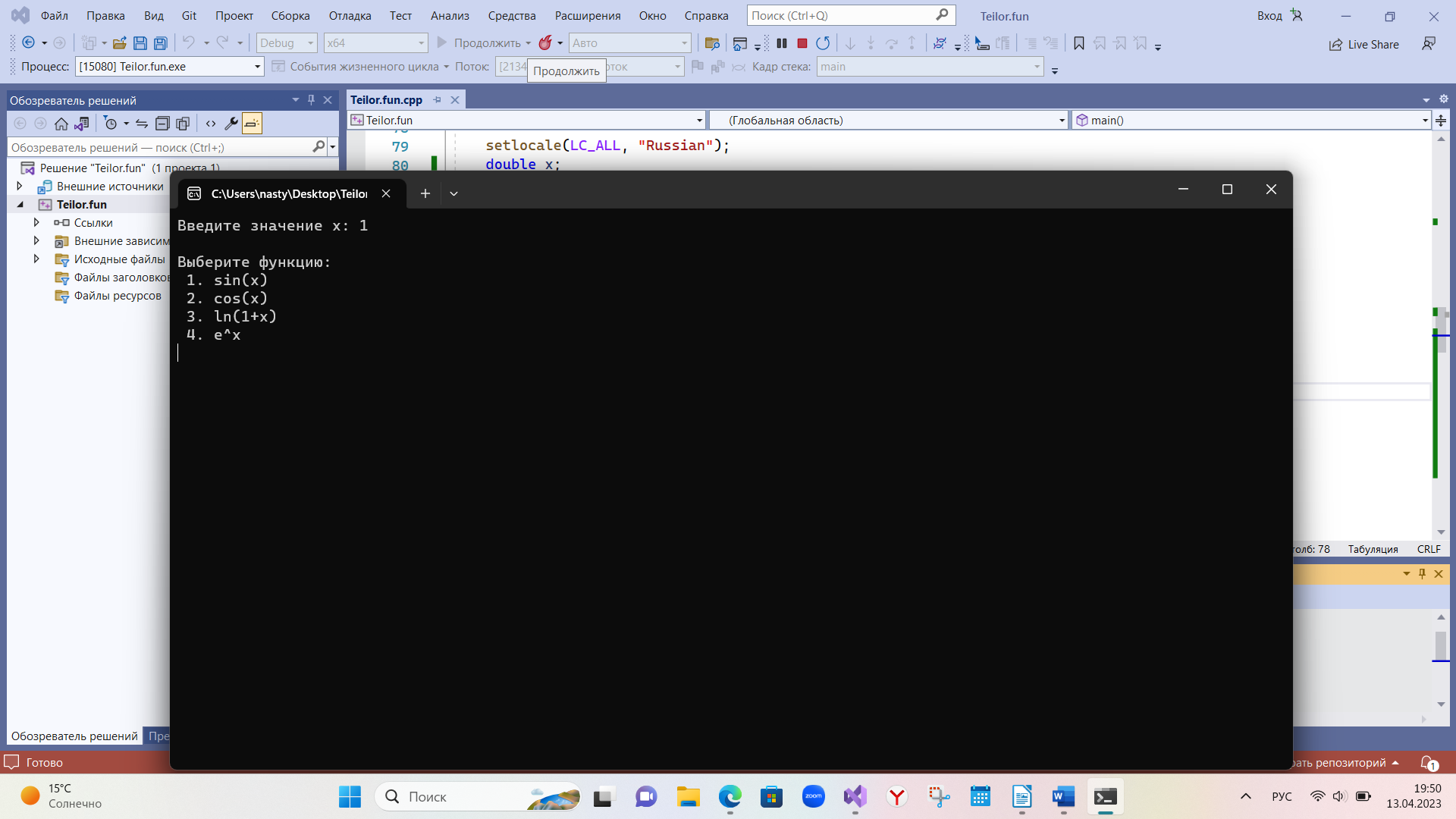


Рис. 2 «Главное меню 2»

После программа подсчитывает значение функции тремя способами: прямой суммой, обратной суммой и встроенной функцией. Также программа вычислит результат абсолютной погрешности вычисления математической функции. Все подсчёты будут выведены на экран.

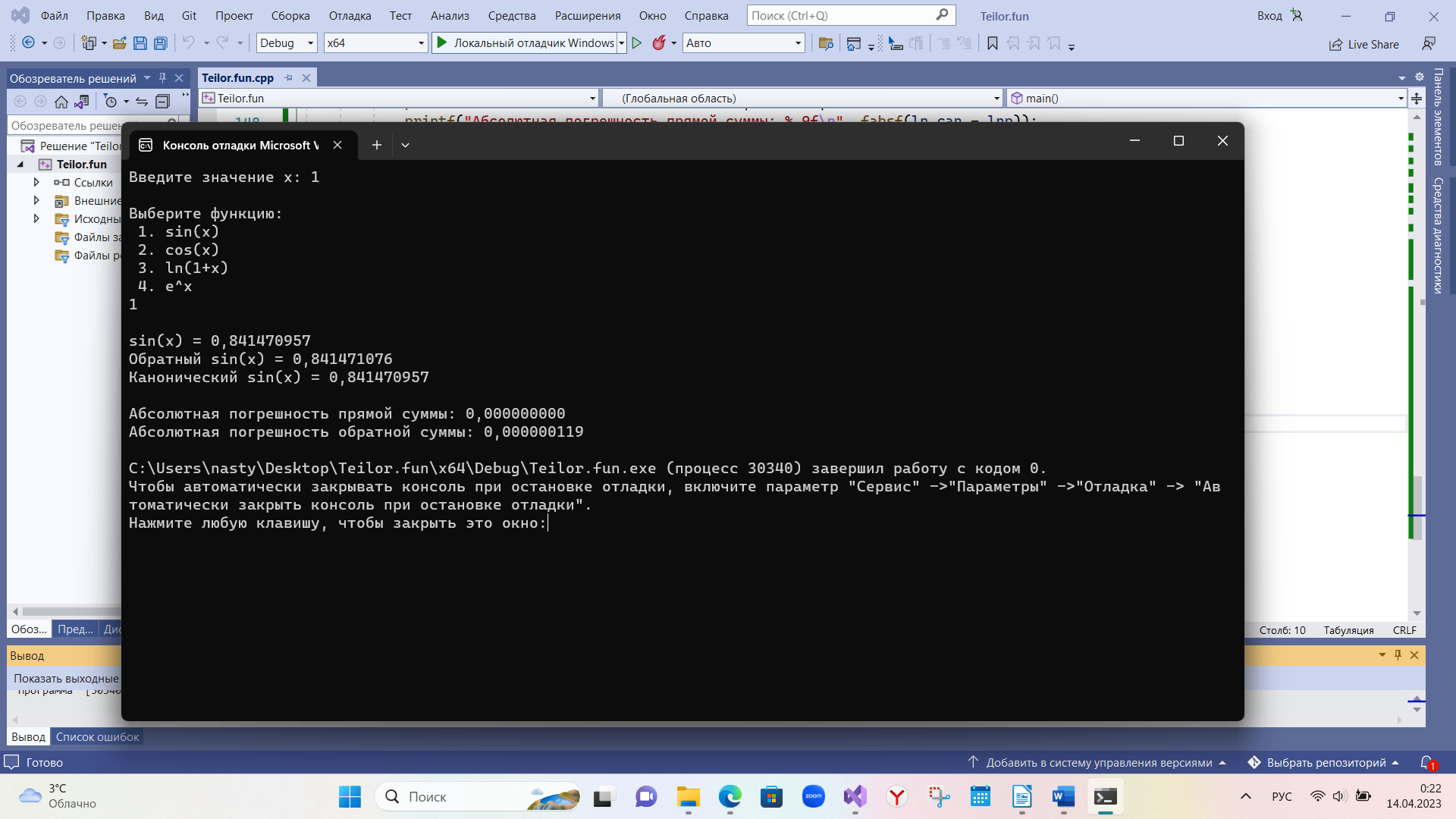


Рис. 3 «Результат работы программы 1»

Если ввести значение х > 1, то для функции ln(1+x) программа выдаст ошибку: «Недопустимые значения для ln(1+x)».

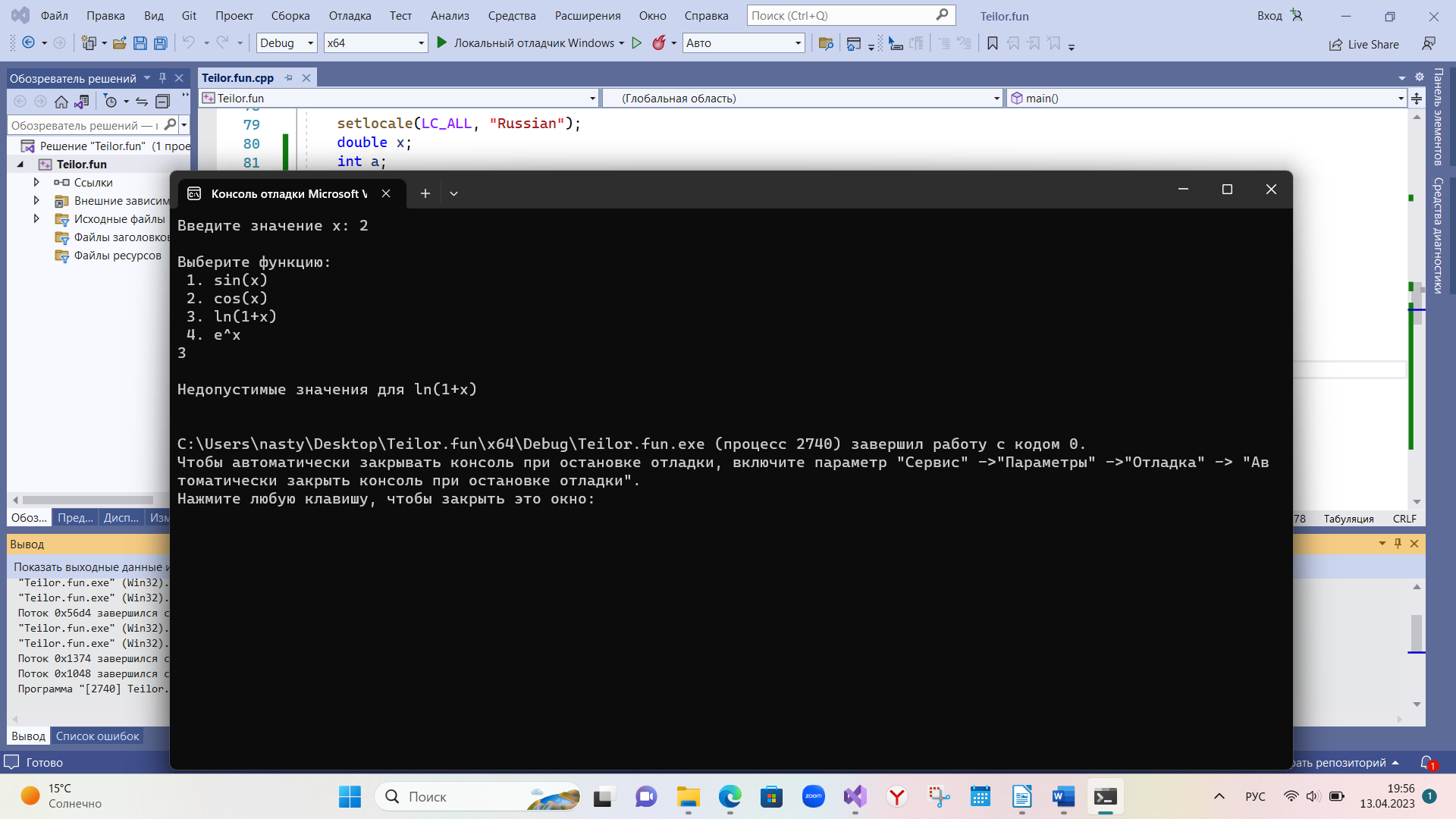


Рис. 4 «Результат работы программы 2»

# Описание программной реализации

Функция **float sin\_pr(float x, float fel, int ind)** возвращает ai-й элемент из вычисления sin(x) по формуле Тейлора при прямой сумме по рекуррентной формуле:

На вход она принимает значение x, предыдущий элемент fel и индекс элемента ind.

Функция **float cos\_pr(float x, float fel, int ind)** возвращает ai-й элемент из вычисления cos(x) по формуле Тейлора при прямой сумме по рекуррентной формуле:

На вход она принимает значение x, предыдущий элемент fel и индекс элемента ind.

Функция **float log\_pr(float x, float fel, int ind)** возвращает ai-й элемент из вычисления ln(1+x) по формуле Тейлора при прямой сумме по рекуррентной формуле:

На вход она принимает значение x (вычисления работают только для -1 < x <= 1, иначе программа выдаст сообщение о недопустимых значениях х для данной функции), предыдущий элемент fel и индекс элемента ind.

Функция **float exp\_pr(float x, float fel, int ind)** возвращает ai-й элемент из вычисления ex по формуле Тейлора при прямой сумме по рекуррентной формуле:

На вход она принимает значение x, предыдущий элемент fel и индекс элемента ind.

Функция **float sin\_ob(float x, float fel, int ind)** возвращает ai-й элемент из вычисления sin(x) по формуле Тейлора при обратной сумме по рекуррентной формуле:

На вход она принимает значение x, предыдущий элемент fel и индекс элемента ind.

Функция **float cos\_ob(float x, float fel, int ind)** возвращает ai-й элемент из вычисления cos(x) по формуле Тейлора при обратной сумме по рекуррентной формуле:

На вход она принимает значение x, предыдущий элемент fel и индекс элемента ind.

Функция **float log\_ob(float x, float fel, int ind)** возвращает ai-й элемент из вычисления ln(1+x) по формуле Тейлора при обратной сумме по рекуррентной формуле:

На вход она принимает значение x (вычисления работают только для -1 < x <= 1, иначе программа выдаст сообщение о недопустимых значениях х для данной функции), предыдущий элемент fel и индекс элемента ind.

Функция **float exp\_ob(float x, float fel, int ind)** возвращает ai-й элемент из вычисления ex по формуле Тейлора при обратной сумме по рекуррентной формуле:

На вход она принимает значение x, предыдущий элемент fel и индекс элемента ind.

Функция **float summ(float suma, float (\*fun)( float, float, int), float x, float fel)** — функция прямого суммирования ряда Тейлора. На вход принимает переменную суммирования suma, указатель на функцию для вычисления элементов ряда заданной функции, значение x и номер элемента n, с которого начинается прямое суммирование. Возвращает результат переменной suma.

Функция **float summ\_ob(float suma, float(\*fun)(float, float, int), float x, float fel, float n) –** функция обратного суммирования рядя Тейлора. На вход принимает переменную суммирования suma, указатель на функцию для вычисления элементов рядя заданной функции, значение х и номер элемента n, с которого начинается обратное суммирование. Возвращает результат переменной suma.

# Подтверждение корректности

Для проверки корректности реализации всех функций и проведения эксперимента используются встроенные функции sin(x), cos(x), ln(1+x) и exp(x) из стандартной библиотеки math.h. На экран выводятся результаты этих функций как «Каноническая <функция>: ».

# Результаты экспериментов

Суть эксперимента заключается в том, чтобы сравнить точность вычислений функций с каноническими значениями в заданных точках, а также сравнить методы суммирования.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Математическая функция | Точки | Прямое суммирование | Обратное суммирование | Каноническое значение | Абсолютная погрешность прямого суммирования | Абсолютная погрешность обратного суммирования |
| sin(x) | 0,5 | 0,479425520 | 0,479425609 | 0,479425550 | 0,00000003 | 0,00000006 |
| 1 | 0,841470957 | 0,841471076 | 0,841470957 | 0 | 0,000000119 |
| 6 | -0,279417664 | -0,279416561 | -0,279415488 | 0,000002176 | 0,000001073 |
| cos(x) | 0,5 | 0,877582610 | 0,877582788 | 0,877582550 | 0,00000006 | 0,000000238 |
| 1 | 0,540302277 | 0,540302396 | 0,540302277 | 0 | 0,000000119 |
| 6 | 0,960166931 | 0,960170329 | 0,960170269 | 0,000003338 | 0,00000006 |
| ln(1+x) | 0,5 | 0,405465066 | 0,405465126 | 0,405465096 | 0,00000003 | 0,00000003 |
| 1 | 0,688172221 | 0,698073089 | 0,693147182 | 0,004974961 | 0,004925907 |
| 6 | - | - | - | - | - |
| ex | 0,5 | 1,648721337 | 1,648721695 | 1,648721218 | 0,000000119 | 0,000000477 |
| 1 | 2,718281984 | 2,718282700 | 2,718281746 | 0,000000238 | 0,000000954 |
| 6 | 403,428771973 | 403,429046631 | 403,428802490 | 0,000030518 | 0,000244141 |

Полученные данные были занесены в таблицу. По ним можно сделать вывод, что что каждый из методов вычисления математических функций не всегда дает точный результат (практически везде абсолютная погрешность прямого суммирования больше нуля) и прямое суммирование иногда дает более точный результат, чем обратное.

# Заключение

В ходе лабораторной работы было реализовано:

1. вычисление четырёх математических функций тремя способами: по формуле Тейлора прямым, обратным суммированием и встроенной функцией;
2. проверка корректности реализации всех функций;
3. проведение сравнения результатов вычисления функций с каноническими значениями.

# Приложение

float sin\_pr(float x, float fel, int ind) {

fel \*= (-1) \* x \* x / (2 \* ind \* (2 \* ind + 1));

return fel;

}//функция для прямого суммирования

float sin\_ob(float x, float fel, int ind) {

fel = -fel \* (2 \* ind) \* (2 \* ind + 1) / (x \* x);

return fel;

} // функция для обратного суммирования

float summ(float suma, float(\*fun)(float, float, int), float x, float fel) {

suma += fel;

for (int i = 1; i < 100; i++) {

fel = fun(x, fel, i);

suma += fel; }

return suma;

} //прямое суммирование

float summ\_ob(float suma, float(\*fun)(float, float, int), float x, float fel, float n) {

suma += fel;

for (int i = n; i > -1; i--) {

fel = fun(x, fel, i);

suma += fel;}

return suma;

} //обратное суммирование

//sinx

fel = x;

sinp = summ(sinp, sin\_pr, x, fel);

sin\_can = sin(x);

printf("sin(x) = %.16f\n", sinp);

//

lel = 1;

for (j = 1; j < 101; j++) {

lel = sin\_pr(x, fel, j);

if (fabsf(fel) - eps < 0) break;

fel = lel;

}

sino = summ\_ob(sino, sin\_ob, x, lel, j-1);

printf("Обратный sin(x) = %.16f\n", sino);

//

printf("Канонический sin(x) = %.16lf\n\n", sin\_can);

break;

//кейс с вычислением значение синуса