

Институт №3 «Системы управления, информатика и электроэнергетика»

Кафедра № 304 «Вычислительные машины, системы и сети»

Информатика

Отчет по лабораторной работе № 1

Вариант № 7

Выполнила бригада группы МЗО-111Бк-21

Багиров Э. Р.

Нуриев Н. Н.

Проверил Секретарев Виталий Евгеньевич

Москва 2021 г.

Оглавление

| | |
|---|-----------|
| Задание | 3 |
| Диаграмма алгоритма..... | 4 |
| Исходный код программы..... | 5 |
| Алгоритмический вид программы | 6 |
| Некорректные тесты | 7 |
| Корректные тесты | 8 |
| Протокол расчета ожидаемого результата | 11 |
| Вывод..... | 12 |

Задание

Кафедра: 304

Курс: ИНФОРМАТИКА

Задание 1: Вычисление суммы бесконечного числового ряда

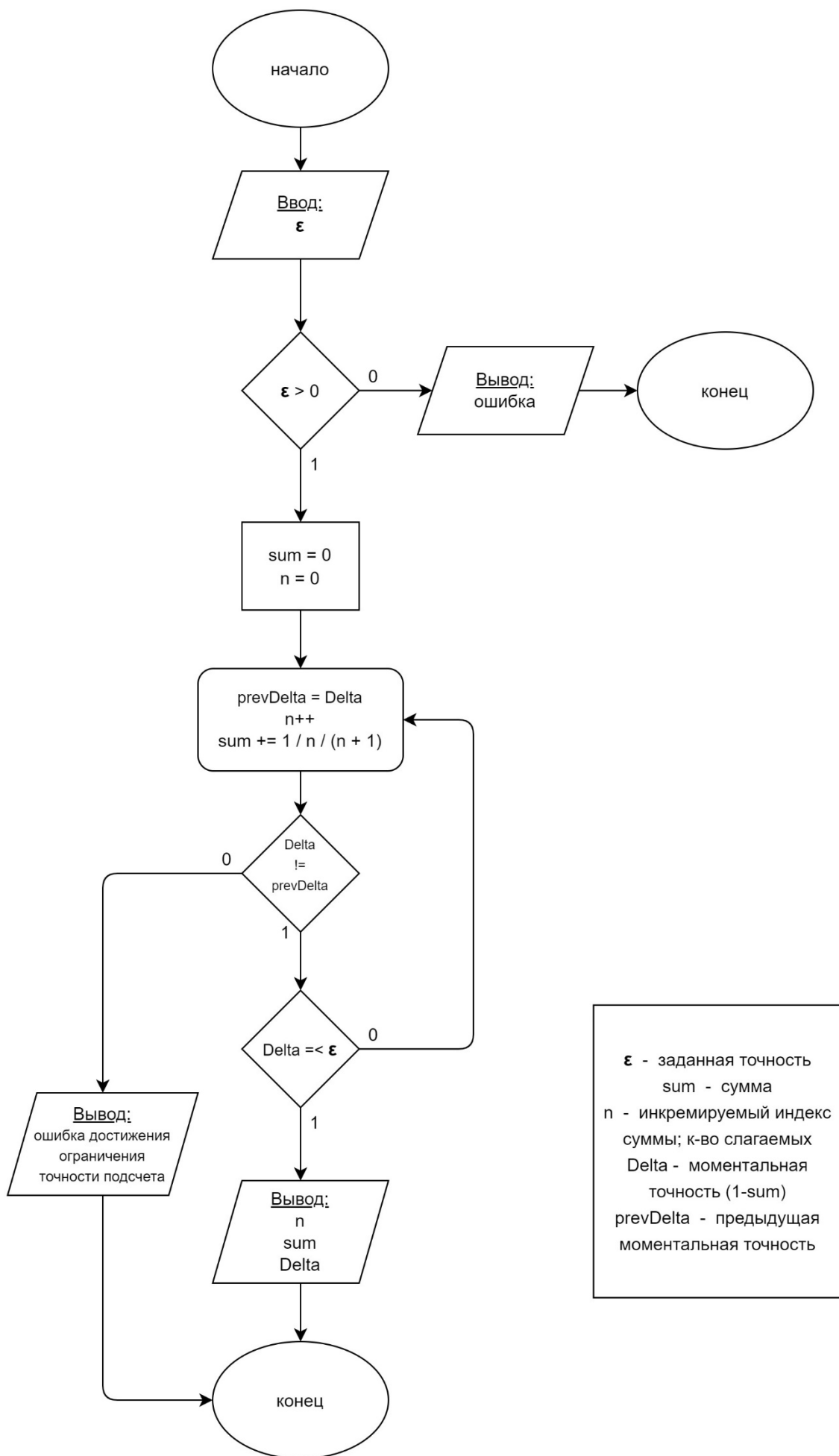
ВАРИАНТ № 7

Определить с заданной точностью сумму бесконечного степенного ряда:

$$\sum_{N=1} \frac{1}{(N)(N+1)} = \frac{1}{1 \times 2} + \frac{1}{2 \times 3} + \frac{1}{3 \times 4} + \dots$$

Предусмотреть ввод точности и печать: количества просуммированных элементов, суммы, разности суммы и точного значения, которое равно: **1**

Диаграмма алгоритма



Исходный код программы

```
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian");

    // Инициализация переменных
    long double Eps;           // заданная пользователем точность подсчета Эпсилон
    double Sum = 0;            // сумма
    unsigned long long N = 0;  // индекс суммы
    double prevDelta = 1;      // моментальная точность предыдущей итерации

    // Входной контроль
    cout << "Введите Eps: ";
    cin >> Eps;
    cout << "\n";
    if (Eps <= 0) {
        // Предотвращение некорректного ввода точности подсчета
        cout << "ОШИБКА - Eps должен быть больше 0." << endl;
        return 0;
    }

    // Расчет суммы
    do {
        N++; // инкремент индекса
        Sum += 1.0 / N / (N+1); // подсчет нового слагаемого, прибавление его к сумме
        if (prevDelta + Sum - 1 == 0) {
            // Если точность не увеличилась за последнюю итерацию,
            // достигнут предел точности, необходимо завершить цикл
            cout << "ОШИБКА - Достигнуто ограничение по точности подсчета.\n";
            return 0;
        }
        prevDelta = 1 - Sum; // сохранение новой моментальной точности
    } while (prevDelta > Eps);

    // Вывод результатов
    cout << "Сумма получена\nКоличество просуммированных элементов N = " << N << endl;
    cout << "Сумма Sum = " << Sum << endl;
    cout << "Достигнутая точность Delta = " << (1 - Sum) << endl;

    return 1;
}
```

Алгоритмический вид программы

АЛГ Вариант №7

ПЕР N, Sum, Eps, interrupted, prevDelta

НАЧАЛО

// Инициализация переменных

Eps // заданная пользователем точность подсчета Эпсилон

Sum = 0 // сумма

N = 0 // индекс суммы

prevDelta = 1 // моментальная точность предыдущей итерации

interrupted = false // флаг о выходе из цикла до достижения необходимой точности

// Входной контроль

ВВОД Eps

ЕСЛИ Eps <= 0

// Предотвращение некорректного ввода точности подсчета

ТО ПЕЧАТЬ (сообщение об ошибке)

КОНЕЦ

КОНЕСЛИ

// Расчет суммы

НЦ ПОВТОРЯТЬ

N = N + 1 // инкремент индекса

Sum += 1.0 / N / (N + 1) // подсчет нового слагаемого, прибавление его к сумме

ЕСЛИ (prevDelta + Sum - 1 == 0)

// Если точность не увеличилась за последнюю итерацию,

// достигнут предел точности, необходимо завершить программу

ТО

ПЕЧАТЬ ошибка достижения ограничения точности

КОНЕЦ

КОНЕСЛИ

prevDelta = 1 - Sum // сохранение новой моментальной точности

КЦ ПОКА (1 - Sum) > Eps

ПЕЧАТЬ (N, Sum, 1 - Sum)

КОНЕЦ

Некорректные тесты

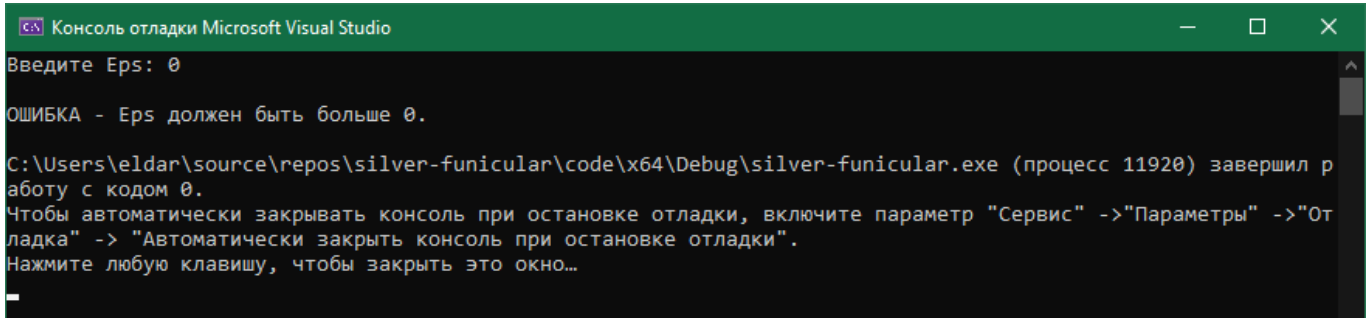
Тест №1

Цель теста: проверить работу программы на границе некорректной области исходных данных

Исходные данные: $Eps = 0$

Ожидаемый результат: вывод сообщения об ошибке: “ОШИБКА - Eps должен быть больше 0.”

Полученный результат:



```
Консоль отладки Microsoft Visual Studio
Введите Eps: 0
ОШИБКА - Eps должен быть больше 0.
C:\Users\eldar\source\repos\silver-funicular\code\x64\Debug\silver-funicular.exe (процесс 11920) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

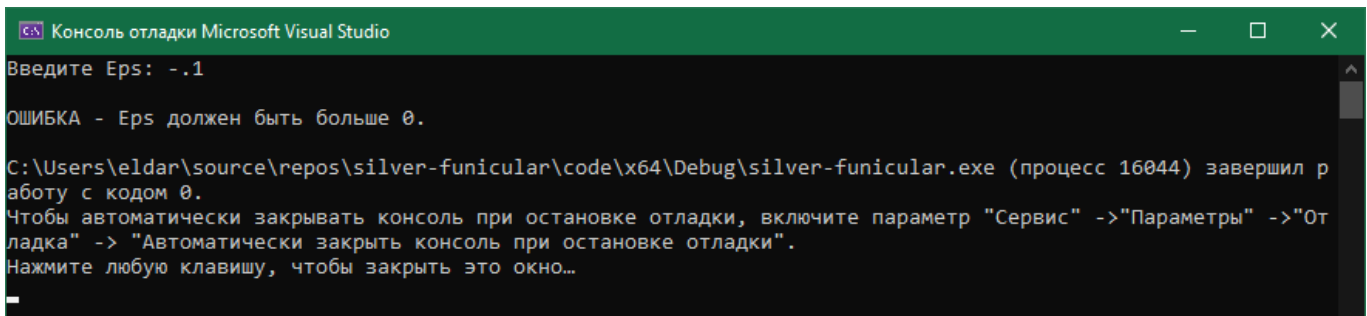
Тест №2

Цель теста: проверить работу программы в некорректной области исходных данных

Исходные данные: $Eps = -0.1$

Ожидаемый результат: вывод сообщения об ошибке: “ОШИБКА - Eps должен быть больше 0.”

Полученный результат:



```
Консоль отладки Microsoft Visual Studio
Введите Eps: -.1
ОШИБКА - Eps должен быть больше 0.
C:\Users\eldar\source\repos\silver-funicular\code\x64\Debug\silver-funicular.exe (процесс 16044) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

Корректные тесты

Тест №1

Цель теста: проверить работу программы в корректной области

Исходные данные: $Eps = 0.5$

Ожидаемый результат: вывод сообщения:

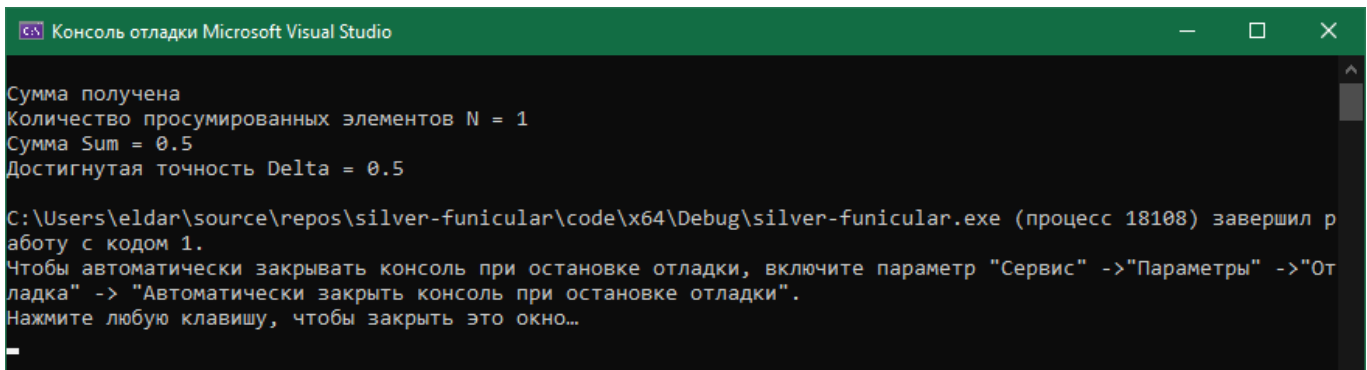
Сумма получена

Количество просуммированных элементов $N = 1$

Сумма $Sum = 0.5$

Достигнутая точность $Delta = 0.5$

Полученный результат:



```
Консоль отладки Microsoft Visual Studio

Сумма получена
Количество просуммированных элементов N = 1
Сумма Sum = 0.5
Достигнутая точность Delta = 0.5

C:\Users\eldar\source\repos\silver-funicular\code\x64\Debug\silver-funicular.exe (процесс 18108) завершил работу с кодом 1.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

Тест №2

Цель теста: проверить работу программы в корректной области

Исходные данные: $Eps = 0.3$

Ожидаемый результат: вывод сообщения:

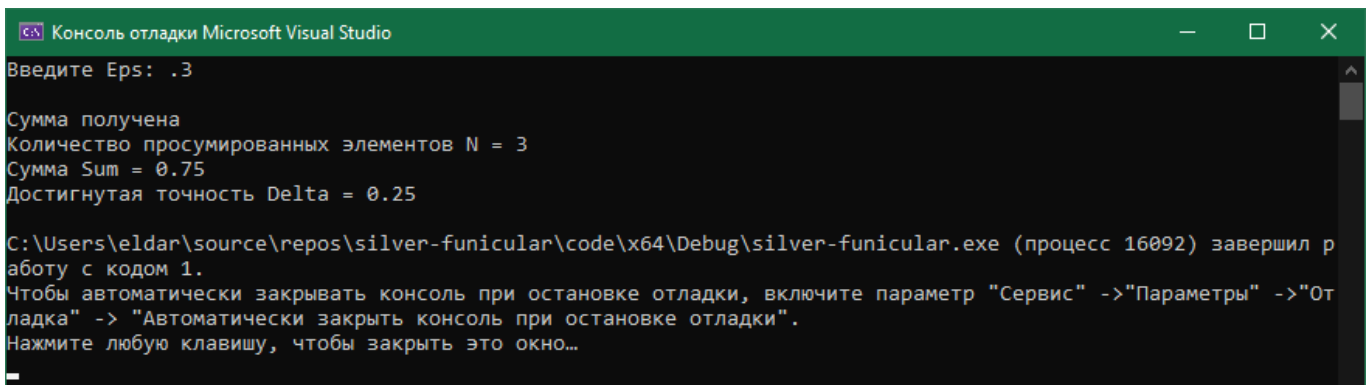
Сумма получена

Количество просуммированных элементов $N = 3$

Сумма $Sum = 0.75$

Достигнутая точность $Delta = 0.25$

Полученный результат:



```
Консоль отладки Microsoft Visual Studio

Введите Eps: .3

Сумма получена
Количество просуммированных элементов N = 3
Сумма Sum = 0.75
Достигнутая точность Delta = 0.25

C:\Users\eldar\source\repos\silver-funicular\code\x64\Debug\silver-funicular.exe (процесс 16092) завершил работу с кодом 1.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

Тест №3

Цель теста: проверить работу программы в корректной области

Исходные данные: $Eps = 0.2$

Ожидаемый результат: вывод сообщения:

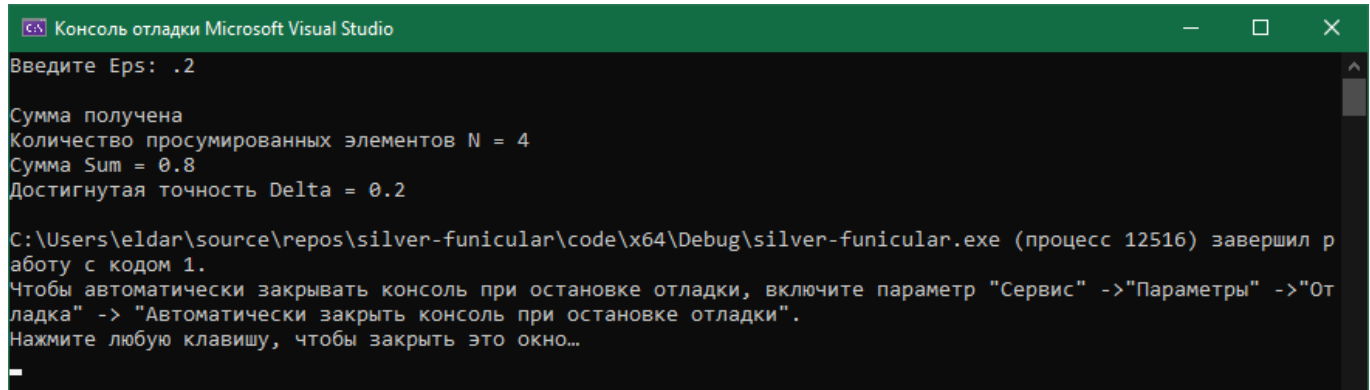
Сумма получена

Количество просуммированных элементов $N = 4$

Сумма $Sum = 0.8$

Достигнутая точность $Delta = 0.2$

Полученный результат:



```
Консоль отладки Microsoft Visual Studio
Введите Eps: .2

Сумма получена
Количество просуммированных элементов N = 4
Сумма Sum = 0.8
Достигнутая точность Delta = 0.2

C:\Users\eldar\source\repos\silver-funicular\code\x64\Debug\silver-funicular.exe (процесс 12516) завершил работу с кодом 1.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

Тест №4

Цель теста: проверить работу программы в корректной области

Исходные данные: $Eps = 0.15$

Ожидаемый результат: вывод сообщения:

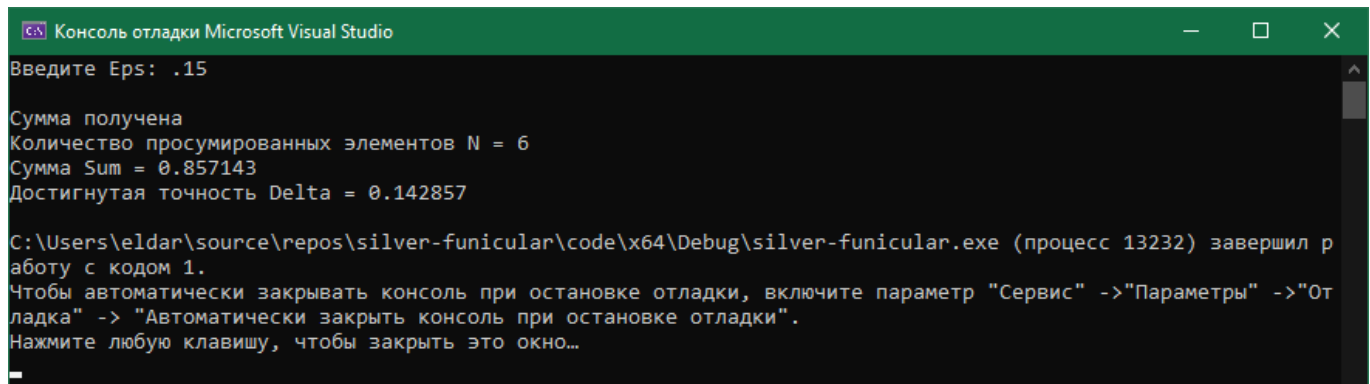
Сумма получена

Количество просуммированных элементов $N = 6$

Сумма $Sum = 0.857143$

Достигнутая точность $Delta = 0.142857$

Полученный результат:



```
Консоль отладки Microsoft Visual Studio
Введите Eps: .15

Сумма получена
Количество просуммированных элементов N = 6
Сумма Sum = 0.857143
Достигнутая точность Delta = 0.142857

C:\Users\eldar\source\repos\silver-funicular\code\x64\Debug\silver-funicular.exe (процесс 13232) завершил работу с кодом 1.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

Тест №5

Цель теста: проверить работу программы в корректной области

Исходные данные: $Eps = 0.1$

Ожидаемый результат: вывод сообщения об ошибке:

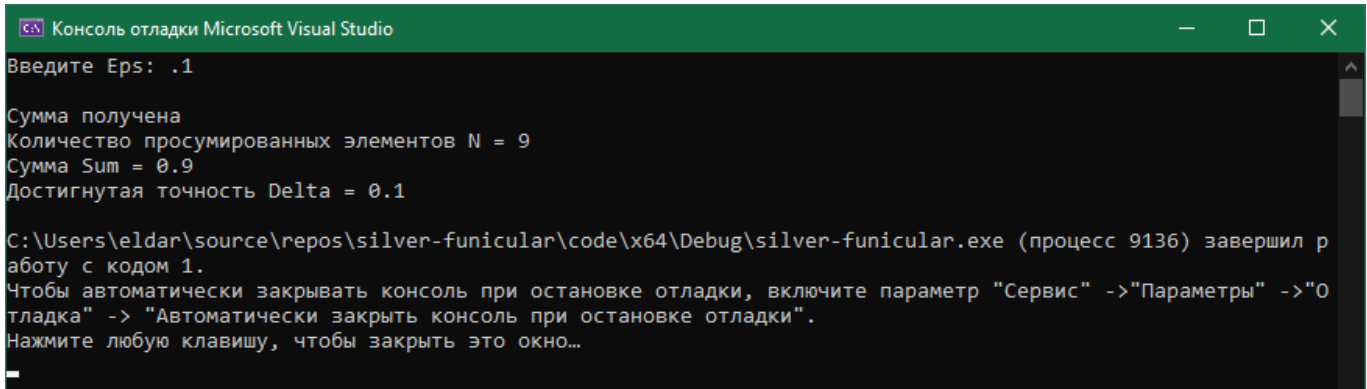
Сумма получена

Количество просуммированных элементов $N = 9$

Сумма $Sum = 0.9$

Достигнутая точность $Delta = 0.1$

Полученный результат:



```
Консоль отладки Microsoft Visual Studio
Введите Eps: .1

Сумма получена
Количество просуммированных элементов N = 9
Сумма Sum = 0.9
Достигнутая точность Delta = 0.1

C:\Users\eldar\source\repos\silver-funicular\code\x64\Debug\silver-funicular.exe (процесс 9136) завершил работу с кодом 1.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

Тест №6

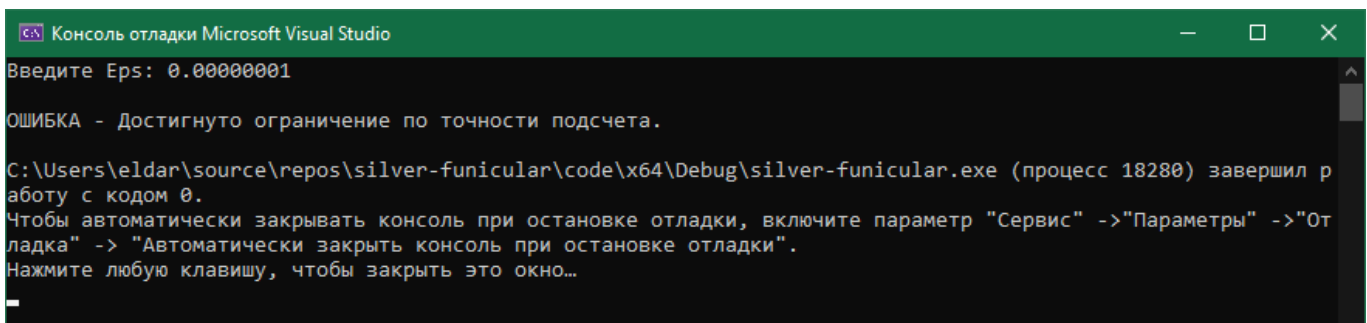
Цель теста: проверить работу программы в корректной области

Исходные данные: $Eps = 0.00000001$

Ожидаемый результат: вывод сообщения об ошибке:

ОШИБКА - Достигнуто ограничение по точности подсчета.

Полученный результат:



```
Консоль отладки Microsoft Visual Studio
Введите Eps: 0.00000001

ОШИБКА - Достигнуто ограничение по точности подсчета.

C:\Users\eldar\source\repos\silver-funicular\code\x64\Debug\silver-funicular.exe (процесс 18280) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Вывод: полученный результат совпал с ожидаемым. Тест ошибки не обнаружил.

Протокол расчета ожидаемого результата

| N | Формула | Sum | Delta (Eps) |
|----|--|-----------|-------------|
| 1 | $\frac{1}{2}$ | .5 | .5 |
| 2 | $\frac{1}{2} + \frac{1}{6}$ | .(6) | .(3) |
| 3 | $\frac{1}{2} + \frac{1}{6} + \frac{1}{12}$ | .75 | .25 |
| 4 | $\frac{1}{2} + \frac{1}{6} + \frac{1}{12} + \frac{1}{20}$ | .8 | .2 |
| 5 | $\frac{1}{2} + \frac{1}{6} + \frac{1}{12} + \frac{1}{20} + \frac{1}{30}$ | .8(3) | .1(6) |
| 6 | $\frac{1}{2} + \frac{1}{6} + \frac{1}{12} + \frac{1}{20} + \frac{1}{30} + \frac{1}{42}$ | .(857142) | .(142857) |
| 7 | $\frac{1}{2} + \frac{1}{6} + \frac{1}{12} + \frac{1}{20} + \frac{1}{30} + \frac{1}{42} + \frac{1}{56}$ | .875 | .125 |
| 8 | $\frac{1}{2} + \frac{1}{6} + \frac{1}{12} + \frac{1}{20} + \frac{1}{30} + \frac{1}{42} + \frac{1}{56} + \frac{1}{72}$ | .(8) | .(1) |
| 9 | $\frac{1}{2} + \frac{1}{6} + \frac{1}{12} + \frac{1}{20} + \frac{1}{30} + \frac{1}{42} + \frac{1}{56} + \frac{1}{72} + \frac{1}{90}$ | .9 | .1 |
| 10 | $\frac{1}{2} + \frac{1}{6} + \frac{1}{12} + \frac{1}{20} + \frac{1}{30} + \frac{1}{42} + \frac{1}{56} + \frac{1}{72} + \frac{1}{90} + \frac{1}{110}$ | .(90) | .(09) |

Вывод

Проанализировали алгоритм по вычислению суммы бесконечного числового ряда. В ходе выполнения лабораторной работы были составлены диаграмма алгоритма и алгоритмический текст программы для дальнейшего написания программы. Проведены некорректные тесты для проверки соответствия исходных данных и корректные тесты, которые показали правильность выполнения алгоритма.