

## ВВЕДЕНИЕ

Цель – данной лабораторной работы – разработка программы для вычисления суммы бесконечного ряда вида

$y = \frac{1}{2^2} + \frac{1}{5^2} + \frac{1}{8^2} + \dots + \frac{1}{(3n-2)^2}$ . Программа должна находить сумму для конечного числа

элементов и выводить результат

Задачи:

- Составить IDEF0-диаграмму проекта;
- Составить блок-схемы алгоритмов;
- Реализовать алгоритмы на языке Object Pascal;
- Протестировать реализации алгоритмов.

## Аналитическая часть

Структура представлена на рисунке 1 в виде IDEF0-нотации. Задача разделяется на несколько блоков, представленных на рисунке 2:

- Ввод данных с клавиатуры;
- Заполнение массива случайными числами;
- Сортировка массива по убыванию;
- Подсчет уникальных элементов;
- Разделение на четные и нечетные элементы.
- Вывод данных на экран.

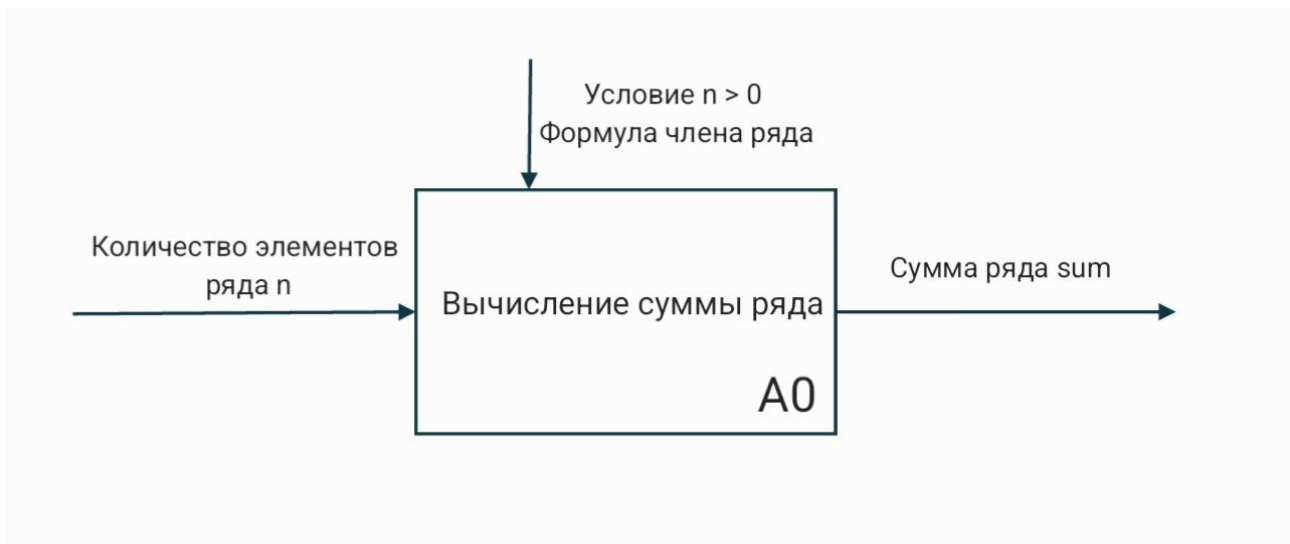


Рисунок 1 – Общая IDEF0-нотация

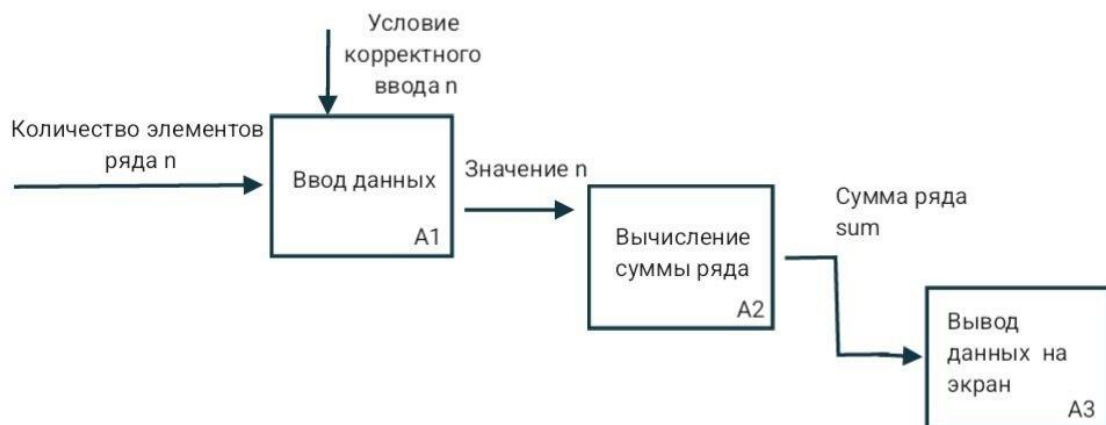


Рисунок 2 – Подробная IDEF0-нотация

## Конструкторская часть

Блоки IDEF0-диаграммы представляют собой 1-2 действия, ввиду чего рациональнее отобразить алгоритм всего проекта целиком без разбиения каждого блока на отдельные процедуры.

Блок-схема алгоритма представлена на рисунке 1. Элементы, отвечающие за интерфейс пользователя, на блок-схеме не отображены; текстовые сообщения, ввиду малозначимости их дословного приведения, представлены сокращенно.

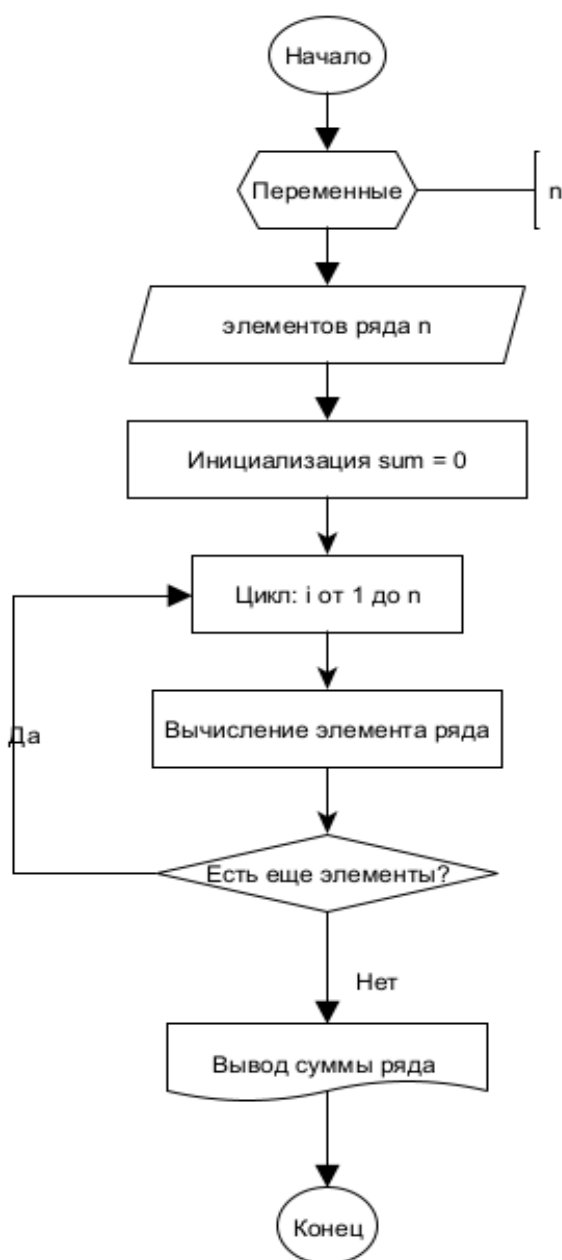


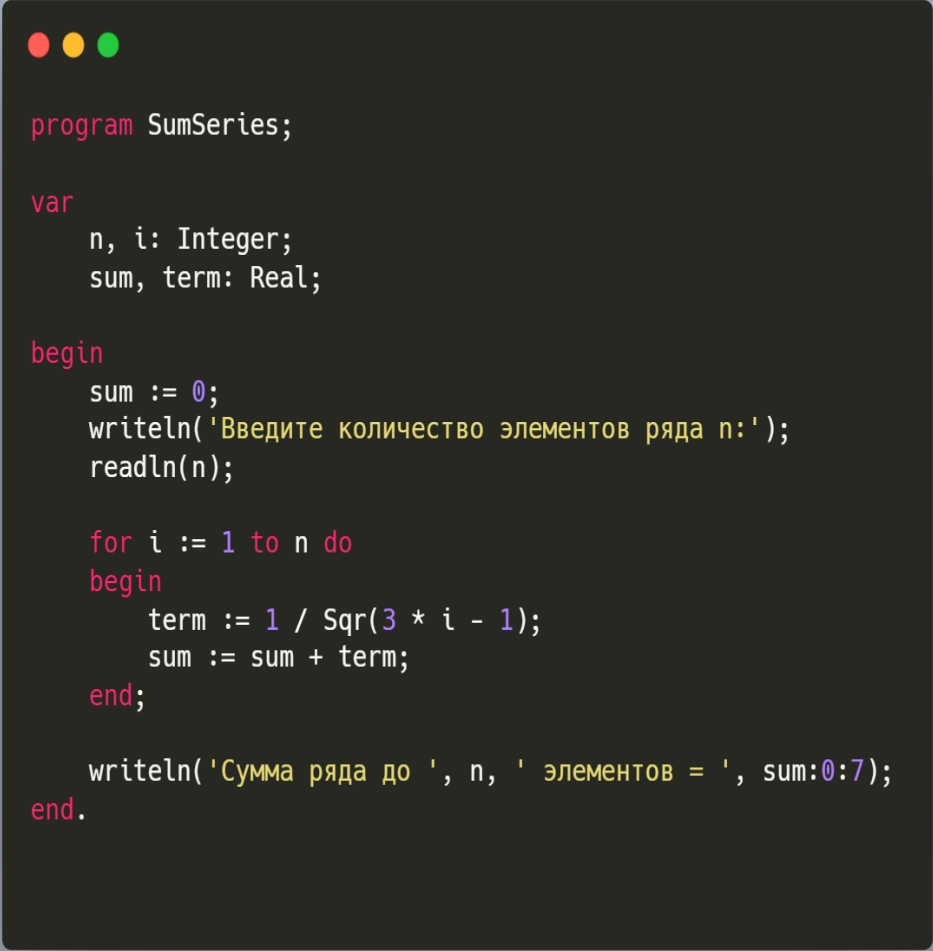
Рисунок 1 – Блок-схема алгоритма программы

## Технологическая часть

### Реализация алгоритма

В настоящем разделе представлена реализация алгоритма, чья блок-схема представлена на рисунке 1. Реализация была произведена с помощью языка программирования Pascal и представлена в листинге 1.

Листинг 1 – Программа алгоритма



```
program SumSeries;

var
  n, i: Integer;
  sum, term: Real;

begin
  sum := 0;
  writeln('Введите количество элементов ряда n:');
  readln(n);

  for i := 1 to n do
  begin
    term := 1 / Sqr(3 * i - 1);
    sum := sum + term;
  end;

  writeln('Сумма ряда до ', n, ' элементов = ', sum:0:7);
end.
```

## Тестирование реализации

Для данной реализации были использованы методы тестирования черным ящиком – метод эквивалентного разбиения – и белым ящиком – метод комбинаторного покрытия условий и решений.

Были выделены следующие эквивалентные классы возможных решений:

1. Корректный ввод количества элементов ряда  $n > 0$  : программа правильно вычисляет сумму ряда до заданного количества элементов.
2. Неверный ввод количества элементов ряда  $n \leq 0$ : программа корректно выводит сообщение об ошибке или не выполняет расчёты.
3. Пограничные значения ввода: программа корректно обрабатывает значения, близкие к нулю, или очень большие значения  $n$ , если они допустимы по ограничениям типа данных.
4. Ошибки ввода: ввод символов, строк или данных, не соответствующих целому числу, приводит к системной ошибке или завершению работы программы.

Результаты тестов представлены в таблице 1.

Таблица 1 – Тесты реализации программы

№	Входные данные	Выходные данные
1	$n=5$	Сумма ряда = 0.3189915
2	$n=10$	Сумма ряда = 0.3295104
3	$n=20$	Сумма ряда = 0.3349221
4	$n=-3$	Сумма ряда = 0.0000000
5	$n='abc'$	Системная ошибка (введено не число)

## ЗАКЛЮЧЕНИЕ

В настоящей работе была разработана программа для вычисления суммы ряда  $y = \frac{1}{2^2} + \frac{1}{5^2} + \frac{1}{8^2} + \dots + \frac{1}{(3n-2)^2}$  проведены её тестирование и анализ.

Задачи:

- Составлена IDEF0-диаграмма проекта;
- Составлены блок-схемы алгоритмов;
- Алгоритмы реализованы на языке Object Pascal;
- Реализации алгоритмов протестированы, были подобраны классы ошибок.