Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ

Дан объект следующей конструкции:

В закрытом доступе имеется массив целого типа и поле его длины. Количество элементов массива четное и больше двух. Объект имеет функциональность:

- Конструктор по умолчанию, в начале работы выдает сообщение;
- Параметризированный конструктор, передается целочисленный параметр. По значению параметра определяется размерность целочисленного массива из закрытой области. Массив создается. В начале работы выдает сообщение;
- Метод деструктор, который выдает сообщение что он отработал;
- Метод ввода данных для созданного массива;
- Метод 1, который суммирует значения очередной пары элементов и сумму присваивает первому элементу пары. Далее суммирует элементы полученного массива и возвращает это значение. Например, пусть массив состоит из элементов {1,2,3,4}. В результате суммирования пар получим массив {3,2,7,4};
- Метод 2, который умножает значения очередной пары элементов и результат присваивает первому элементу пары. Далее суммирует элементы полученного массива и возвращает это значение. Например, пусть массив состоит из элементов {1,2,3,4}. В результате умножения пар получим массив {2,2,12,4};
- Метод который, суммирует значения элементов массива и возвращает это значение.

Разработать функцию, которая в качестве параметра получает объект по значению. Функция вызывается метод 2, далее выводит сумму элементов массива с новой строки.

В основной функции реализовать алгоритм:

1. Ввод размерности массива. Размер должен иметь значение больше 2 и быть четным.
2. Если размерность массива некорректная, вывод сообщения и завершить работу алгоритма.
3. Вывод значения размерности массива.
4. Создание объекта с аргументом размерности массива.
5. Вызов метода для ввода значений элементов массива.
6. Вызов функции передача в качестве аргумента объекта.
7. Вызов метода 1 от имени объекта.
8. Вывод суммы элементов массива объекта с новой строки.

Разработать конструктор копии объекта для корректного выполнения вычислений. В начале работы конструктор копии выдает сообщение с новой строки.

1.1 Описание входных данных

Первая строка:

«Целое число»

Вторая строка:

«Целое число» «Целое число» . . .

Пример:

```
8
1 2 3 4 5 6 7 8
```

1.2 Описание выходных данных

Если введенная размерность массива допустима, то в первой строке выводится это значение:

```
«Целое число»
```

Если введенная размерность массива не больше двух или нечетная, то в первой строке выводится некорректное значение и вопросительный знак:

```
«Целое число»?
```

Конструктор по умолчанию в начале работы с новой строки выдает сообщение:

```
Default constructor
```

Параметризированный конструктор в начале работы с новой строки выдает сообщение:

```
Constructor set
```

Конструктор копирования в начале работы с новой строки выдает сообщение:

```
Copy constructor
```

Деструктор в начале работы с новой строки выдает сообщение:

```
Destructor
```

Пример вывода:

```
8
Constructor set
Copy constructor
120
Destructor
56
Destructor
```

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- оператор ввода и вывода.

3 ОПИСАНИЕ АЛГОРИТМОВ

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

3.1 Алгоритм функции main

Функционал: осовная функция.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		int razmer	2
2		ввод razmer	3
3	razmer <= 2 || razmer %2 != 0	вывод ошибок	∅
			4
4		вывод razmer	5
5		создание объека obj	6
6		вывод vod	7
7		вызов fun	8
8		вызов second	9
9		вызов summa	∅

3.2 Алгоритм функции fun

Функционал: копирование.

Параметры: нет.

Возвращаемое значение: отсутствует.

Алгоритм функции представлен в таблице 2.

Таблица 2 – Алгоритм функции fun

№	Предикат	Действия	№ перехода
1		вызов first	2
2		вызов summa	∅

3.3 Алгоритм конструктора класса Class

Функционал: создание объекта.

Параметры: нет.

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса Class

№	Предикат	Действия	№ перехода
1		вывод Default constructor	∅

3.4 Алгоритм конструктора класса Class

Функционал: создаие объекта.

Параметры: razmer.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса Class

№	Предикат	Действия	№ перехода
1		вывод Constructor set	2
2		создать цклочисленный массив razmer	∅

3.5 Алгоритм конструктора класса Class

Функционал: создание объекта.

Параметры: Class& obj.

Алгоритм конструктора представлен в таблице 5.

Таблица 5 – Алгоритм конструктора класса Class

№	Предикат	Действия	№ перехода
1		вывод Copy constructor	2
2		копирование obj	∅

3.6 Алгоритм деструктора класса Class

Функционал: удалить объект.

Параметры: нет.

Алгоритм деструктора представлен в таблице 6.

Таблица 6 – Алгоритм деструктора класса Class

№	Предикат	Действия	№ перехода
1		вывод Destructor	∅

3.7 Алгоритм метода vod класса Class

Функционал: ввод значений.

Параметры: нет.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода vod класса Class

№	Предикат	Действия	№ перехода
1		int a	2
2	i < razmer		3
			∅
3		ввод a	4
4		arr[i]=a	5
5		инкремент i	∅

3.8 Алгоритм метода first класса Class

Функционал: сложение элемента массива.

Параметры: нет.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода first класса Class

№	Предикат	Действия	№ перехода
1	i < razmer		2
			∅
2		arr[i]=arr[i]+arr[i+1]	3
3		инкремен i	1

3.9 Алгоритм метода second класса Class

Функционал: умножение элементов массива.

Параметры: нет.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода second класса Class

№	Предикат	Действия	№ перехода
1	i < razmer		2
			∅
2		arr[i]=arr[i]*arr[i+1]	3
3		инкремент i	∅

3.10 Алгоритм метода summa класса Class

Функционал: суммирлвание.

Параметры: нет.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 10.

Таблица 10 – Алгоритм метода summa класса Class

№	Предикат	Действия	№ перехода
1		int summa	2
2	i < razmer		3
			5
3		summa = summa + arr[i]	4
4		инкремент i	2
5		вывод summa	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-7.

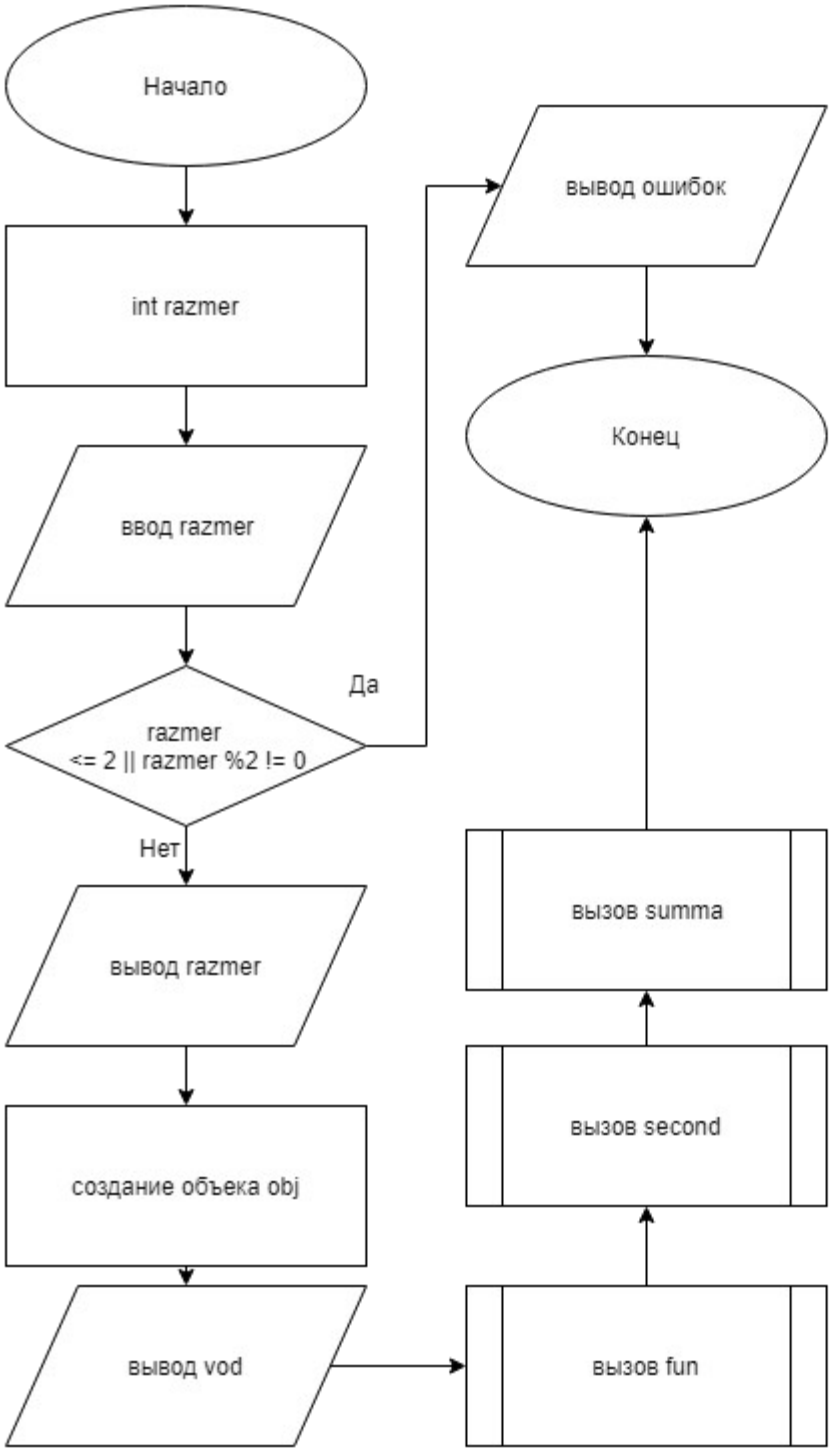

Рисунок 1 – Блок-схема алгоритма

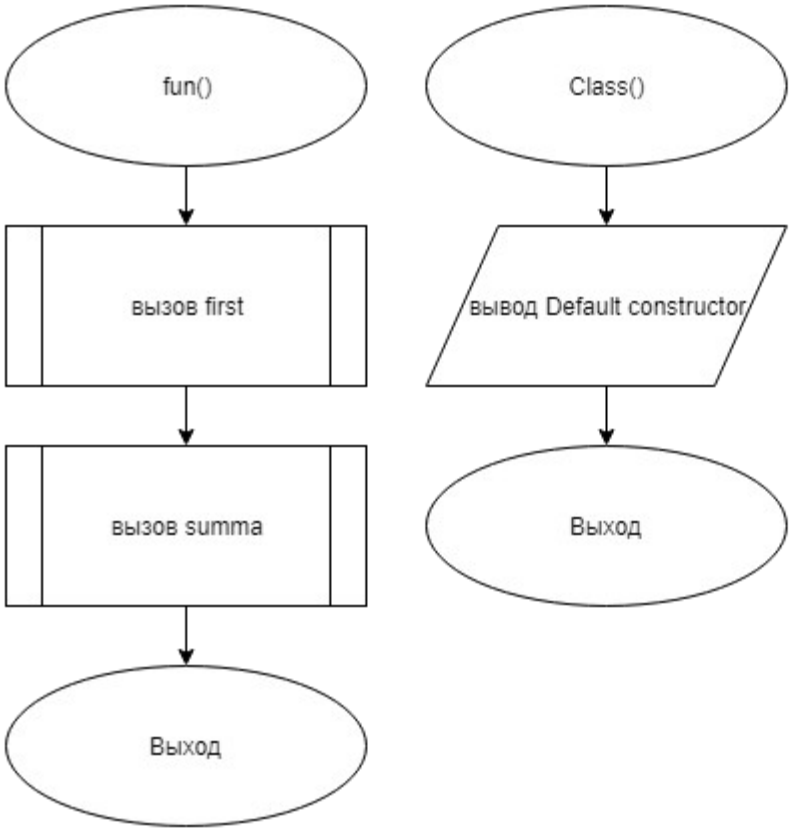

Рисунок 2 – Блок-схема алгоритма

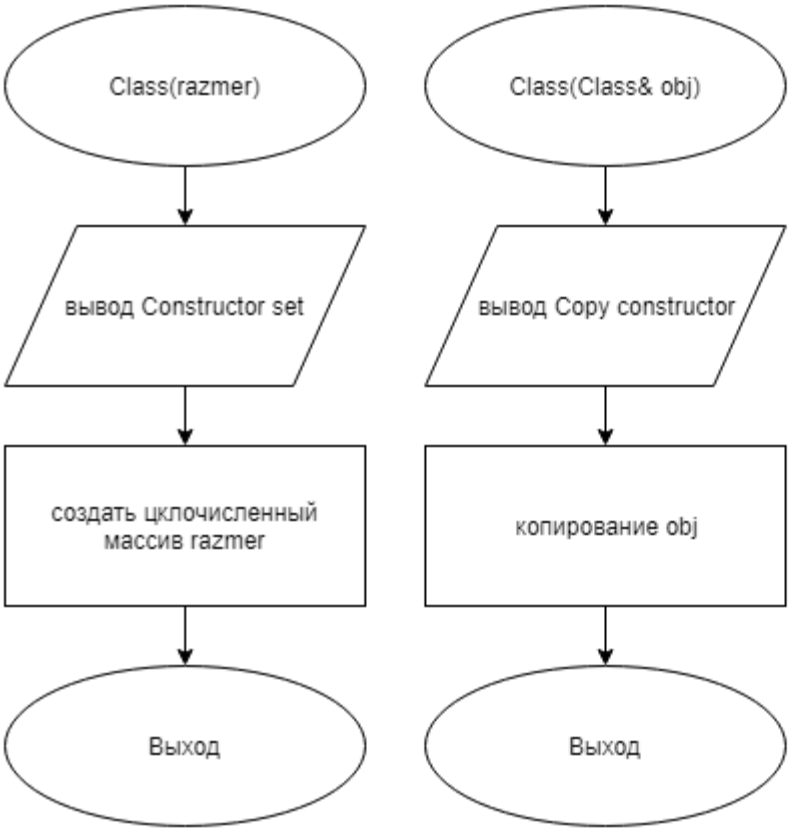

Рисунок 3 – Блок-схема алгоритма

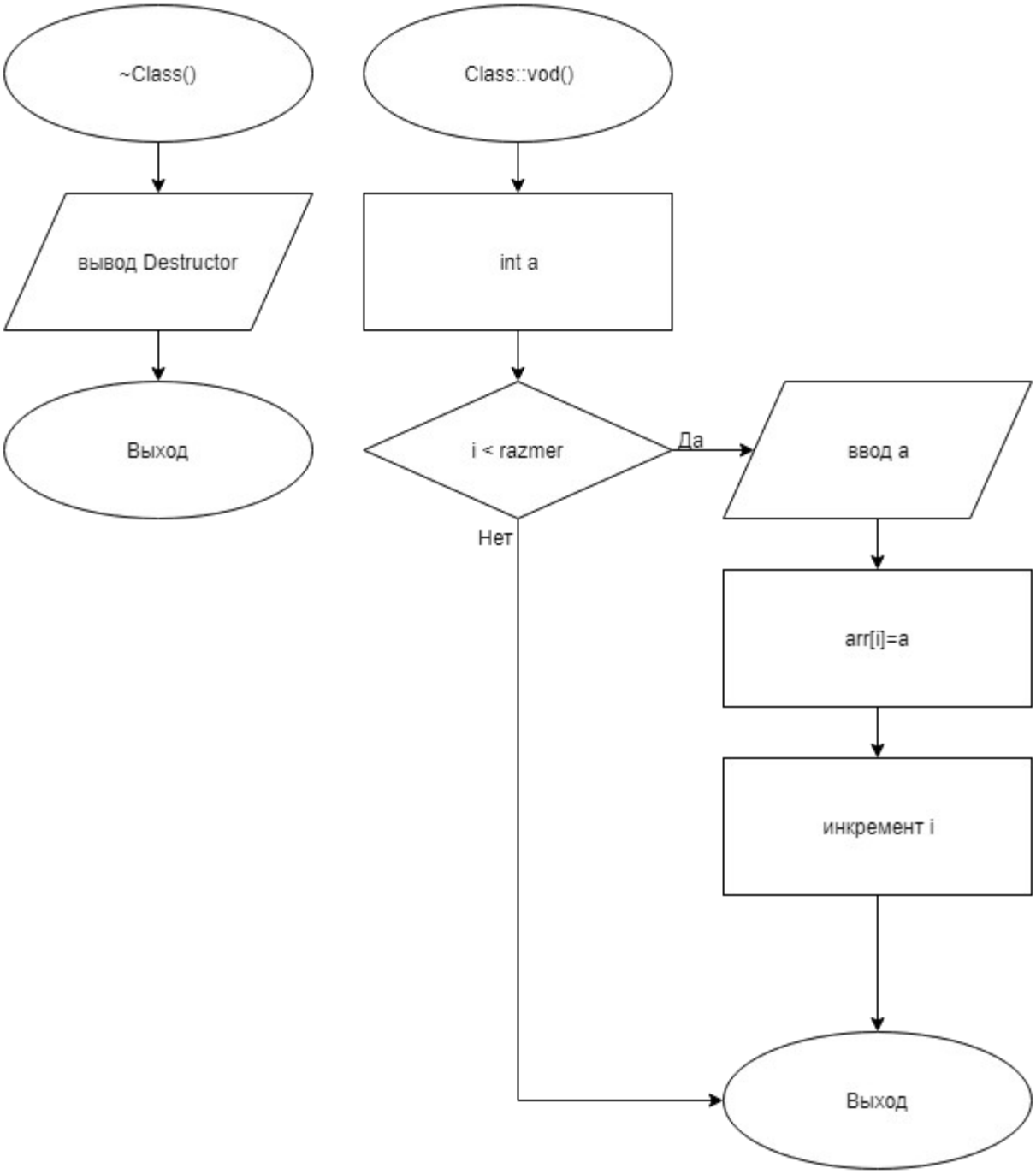

Рисунок 4 – Блок-схема алгоритма

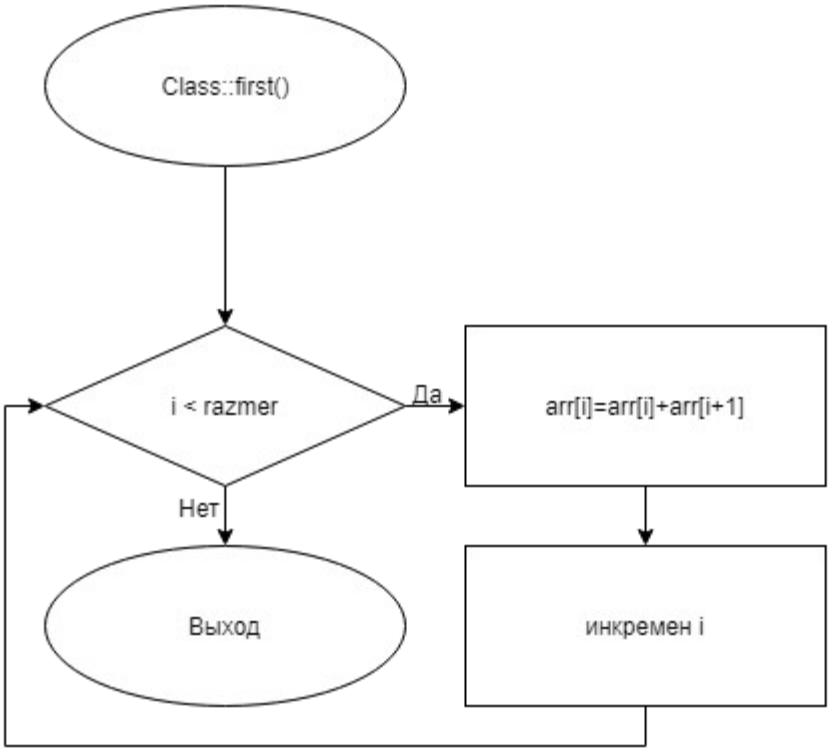

Рисунок 5 – Блок-схема алгоритма

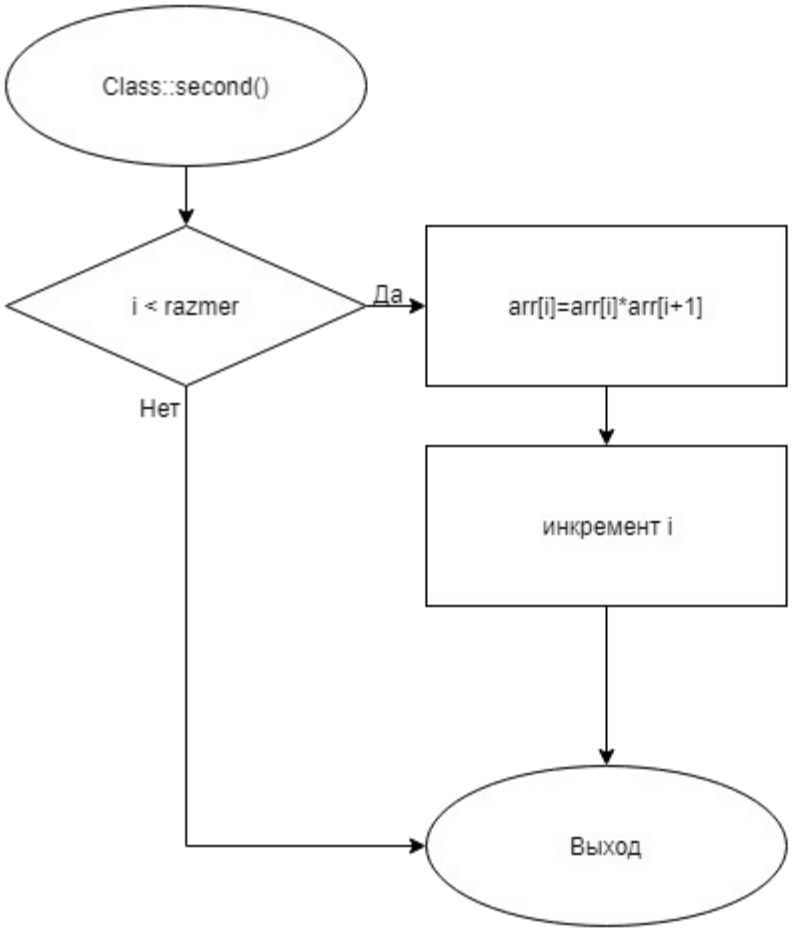

Рисунок 6 – Блок-схема алгоритма

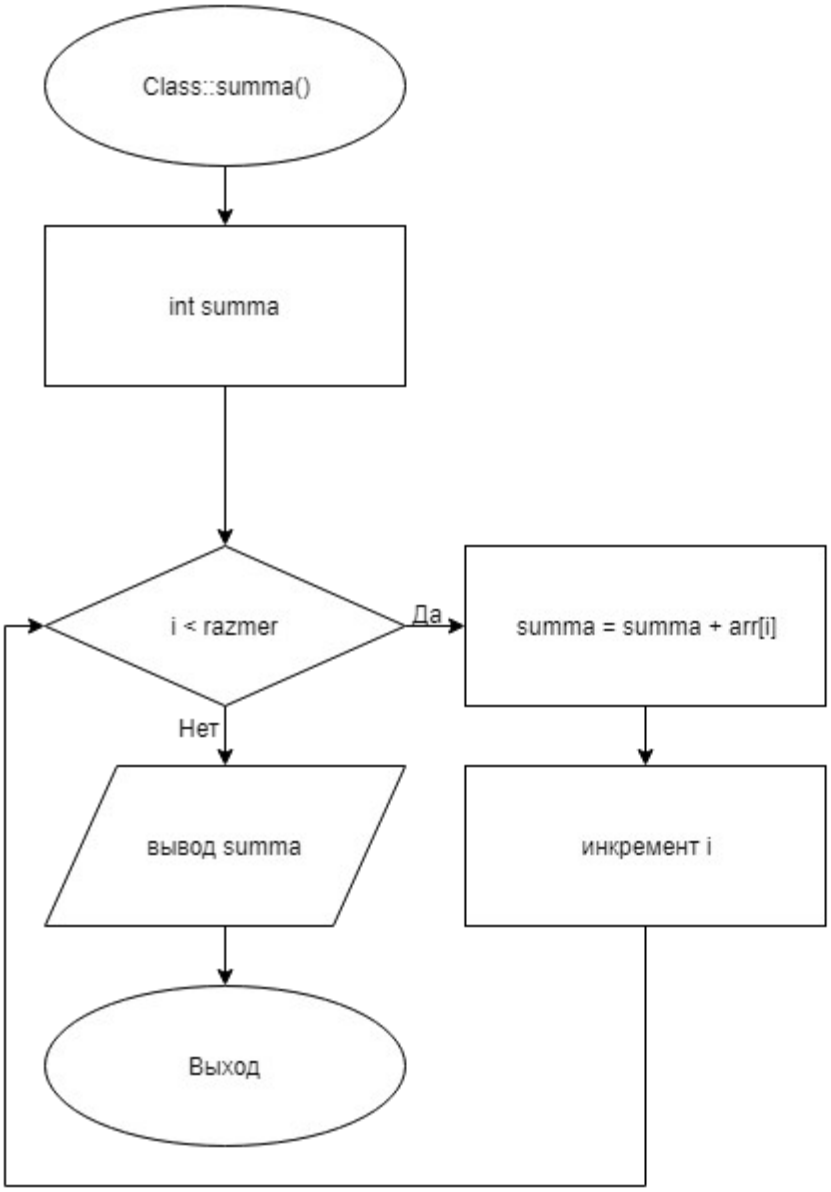

Рисунок 7 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл Class.cpp

Листинг 1 – Class.cpp

```
#include "Class.h"
#include <iostream>
using namespace std;

Class::Class()
{
   cout << "Default constructor" << endl;
}

Class::Class(int razmer)
{
   arr = new int [razmer];
   cout <<"Constructor set"<< endl;
   this->razmer = razmer;
}

Class::Class(const Class & obj)
{
   cout <<"Copy constructor"<< endl;
   razmer=obj.razmer;
   arr = new int [razmer];
   for (int i = 0; i < razmer; i++)
   {
      arr[i] = obj.arr[i];
   }
}
int Class::vod()
{
   int a;
   for (int i = 0; i < razmer; i++)
   {
      cin >> a;
      arr[i] = a;
   }
   return(a);
}
Class::~Class()
{
   cout << "Destructor";
```

```
        if (arr != nullptr)
           delete[]arr;
     }
     int Class::first()
     {
        for (int i = 0; i < razmer; i = i + 2)
        {
           arr[i] = arr[i] * arr[i + 1];
        }
        return(razmer);
     }

     int Class::second()
     {
        for (int i = 0; i < razmer; i = i+2)
        {
           arr[i]=arr[i]+arr[i+1];
        }
        return(razmer);
     }

     void Class::summa()
     {
        int summa = 0;
        for (int i = 0; i < razmer; i++)
        {
           summa = summa + arr[i];
        }
        cout << summa << endl;
     }
```

5.2 Файл Class.h

Листинг 2 – Class.h

```
     #ifndef __CLASS__H
     #define __CLASS__H
     class Class
     {
     private:
        int* arr;
        int razmer;
     public:
        Class();
        Class(int razmer);
        Class(const Class& obj);
        ~Class();
        int vod();
        int first();
        int second();
```

```
   void summa();
};
#endif
```

5.3 Файл main.cpp

Листинг 3 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include "Class.h"
#include <iostream>
using namespace std;

void fun(Class obj)
{
   obj.first();
   obj.summa();
}
int main()
{
   int razmer;
   cin >> razmer;
   if ((razmer <= 2) || (razmer % 2 != 0))
   {
      cout << razmer << "?";
      return 0;
   }
   cout << razmer << endl;
   Class obj(razmer);
   obj.vod();
   fun(obj);
   cout << endl;
   obj.second();
   obj.summa();
   return(0);
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 11.

Таблица 11 – Результат тестирования программы

Входные данные	**Ожидаемые выходные данные**	**Фактические выходные данные**
9 1	9?	9?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19 Единая система программной документации.

2. Методическое пособие студента для выполнения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avrora.ru/student/files/methodichescoe_posobie_dlya_laboratornyh_rabot_3.pdf (дата обращения 05.05.2021).

3. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avrora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).

4. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2019. — 624 с.

5. Видео лекции по курсу «Объектно-ориентированное программирование» [Электронный ресурс]. АСО «Аврора».

6. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие /Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).