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. Параметр должен иметь значение больше 2 и быть четным. По значению параметра определяется размерность целочисленного массива из закрытой области. 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:

- Ввод размерности массива.
- Вывод значения размерности массива.
- Если размерность массива некорректная, вывод сообщения и завершить работу алгоритма.
- Вывод значения размерности массива.
- Создание объекта с аргументом размерности массива.
- Вызов метода для ввода значений элементов массива.
- Вызов функции передача в качестве аргумента объекта.
- Вызов метода 1 от имени объекта.
- 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

Используемые инструменты:

- объект стандартного потока ввода cin (используется для ввода с клавиатуры);
- объект стандартного потока вывода cout (используется для вывода на экран);
- условный оператор (оператор ветвления) if .. else;
- оператор цикла со счётчиком for;
- объект obj класса Object.

Класс Objec:

- Свойства (поля):
  - Целочисленный массив:
    - Наименование - array;
    - Тип - динамический массив целых чисел;
    - Модификатор доступа - закрытый;
  - Размер массива:
    - Наименование - size;
    - Тип - целочисленный;
    - Модификатор доступа - закрытый;
- Методы:
  - Конструктор по умолчанию:
    - Функционал - создание объекта Object с пустым массивом array;
    - Возвращаемое значение - void (отсутствует);
    - Модификатор доступа - открытый;
    - Параметры - отсутствуют;
  - Конструктор с целочисленный параметром:
    - Функционал - создание объекта Object с массивом array размера

size (параметр);

- Возвращаемое значение - void (отсутствует);
- Модификатор доступа - открытый;
- Параметры - целочисленный параметр size;

- Конструктор копирования:
  - Функционал - создание объекта Object со всеми полями равными соответствующим полям obj (параметр);
  - Возвращаемое значение - void (отсутствует);
  - Модификатор доступа - открытый;
  - Параметры - объект класса Object obj;
- Деструктор:
  - Функционал - удаление объекта Object;
  - Возвращаемое значение - void (отсутствует);
  - Модификатор доступа - открытый;
  - Параметры - отсутствуют;
- Метод read_array:
  - Функционал - считывания значений с клавиатуры в массив array;
  - Возвращаемое значение - void (отсутствует);
  - Модификатор доступа - открытый;
  - Параметры - отсутствуют;
- Метод method1:
  - Функционал - суммирование значений очередной пары элементов массива array и присваивание первому элементу пары, далее суммирование значений всех элементов массива и возвращение этого значения;
  - Возвращаемое значение - int (целочисленное значение суммы элементов массива);

- Модификатор доступа - открытый;
- Параметры - отсутствуют;

- Метод method2:
  - Функционал - умножение значений очередной пары элементов массива array и присваивание первому элементу пары, далее суммирование значений всех элементов массива и возвращение этого значения;
  - Возвращаемое значение - int (целочисленное значение суммы элементов массива);
  - Модификатор доступа - открытый;
  - Параметры - отсутствуют;
- Метод sum:
  - Функционал - вычислить сумму элементов массива array;
  - Возвращаемое значение - int (целочисленное значение суммы элементов массива);
  - Модификатор доступа - открытый;
  - Параметры - отсутствуют.

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: основной алгоритм программы.

Параметры: .

Возвращаемое значение: целочисленный код завершения работы программы.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной size значением 0	2
2		Ввод значения переменной size	3
3		Вывод значения переменной size	4
4	Значение переменной size меньше 2 или остаток от деления значения переменной size на 2 равен 1	Вывод "?"	∅
			5
5		Создание объекта obj класса Object с использованием параметризованного конструктора и переменной size в качестве параметра	6
6		Вызов метода read_array объекта obj	7
7		Вызов функции f с объектом obj в качестве	8

№	Предикат	Действия	№ перехода
		аргумента	
8		Вызов метода method1 объекта obj	9
9		Вывод "\n" и возвращённого значения полученого вызовом метода sum объекта obj	∅

3.2 Алгоритм функции f

Функционал: Вызывать method2 объекта, вывести сумму элементов массива объекта.

Параметры: объект класса Object obj.

Возвращаемое значение: void (отсутствует).

Алгоритм функции представлен в таблице 2.

Таблица 2 – Алгоритм функции f

№	Предикат	Действия	№ перехода
1		Вызов метода method2 объекта obj	2
2		Вывод "\n" и возвращённого значения полученого вызовом метода sum объекта obj	∅

3.3 Алгоритм конструктора класса Object

Функционал: создание объекта Object с пустым массивом array.

Параметры: .

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса Object

№	Предикат	Действия	№ перехода
1		Вывод "\nDefault constructor"	2
2		Присваивание полю size значения 0	3

№	Предикат	Действия	№ перехода
3		Присваивание полю array значения nullptr	∅

3.4 Алгоритм конструктора класса Object

Функционал: создание объекта Object с массивом array размера size (параметр);.

Параметры: целочисленный параметр size.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса Object

№	Предикат	Действия	№ перехода
1		Вывод "\nConstructor set"	2
2		Присваивание полю size значения параметра size	3
3		Присваивание полю array адреса нового целочисленного массива размера size созданного с помощью оператора new	∅

3.5 Алгоритм конструктора класса Object

Функционал: создание объекта Object со всеми полями равными соответствующим полям obj (параметр).

Параметры: объект класса Object obj.

Алгоритм конструктора представлен в таблице 5.

Таблица 5 – Алгоритм конструктора класса Object

№	Предикат	Действия	№ перехода
1		Вывод "\nCopy constructor"	2
2		Присваивание полю size значения поля size объекта obj	3
3		Присваивание полю array адреса нового	4

№	Предикат	Действия	№ перехода
		целочисленного массива размера size созданного с помощью оператора new	
4		Инициализация целочисленной переменной i значением 0	5
5	Значение переменной i меньше значения поля size	Присваивание элементу массива (поля) array с индексом i значения элемента массива (поля) array объекта object с индексом i	6
			∅
6		Увеличение значения переменной i на 1	5

3.6 Алгоритм деструктора класса Object

Функционал: удаление объекта Object.

Параметры: .

Алгоритм деструктора представлен в таблице 6.

Таблица 6 – Алгоритм деструктора класса Object

№	Предикат	Действия	№ перехода
1		Вывод "\nDestructor"	2
2		Освобождение памяти по указателю array с помощью оператора delete	∅

3.7 Алгоритм метода read_array класса Object

Функционал: считывания значений с клавиатуры в массив array.

Параметры: .

Возвращаемое значение: void (отсутствует).

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода read_array класса Object

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной i значением 0	2
2	Значение переменной i меньше значения поля size	Ввод элемента массива array с индексом i	3
			∅
3		Увеличение значения переменной i на 1	2

3.8 Алгоритм метода method1 класса Object

Функционал: суммирование значений очередной пары элементов массива array и присваивание первому элементу пары, далее суммирование значений всех элементов массива и возвращение этого значения.

Параметры: .

Возвращаемое значение: int (целочисленное значение суммы элементов массива).

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода method1 класса Object

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной i значением 0	2
2	Значение переменной i меньше результата вычитания 1 из значения поля size	Присваивание i-тому элементу массива array результат суммы значений i-того элемента массива array и (i + 1)-ого элемента массива array	3
			4
3		Увеличение значения переменной i на 2	2

№	Предикат	Действия	№ перехода
4		Возврат значения возвращённого вызванным методом sum	∅

3.9 Алгоритм метода method2 класса Object

Функционал: умножение значений очередной пары элементов массива array и присваивание первому элементу пары, далее суммирование значений всех элементов массива и возвращение этого значения.

Параметры: .

Возвращаемое значение: int (целочисленное значение суммы элементов массива).

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода method2 класса Object

№	Предикат	Действия	№ перехода
1		Инициализация целочисленной переменной i значением 0	2
2	Значение переменной i меньше результата вычитания 1 из значения поля size	Присваивание i-тому элементу массива array результат умножения значений i-того элемента массива array и (i + 1)-ого элемента массива array	3
			4
3		Увеличение значения переменной i на 2	2
4		Возврат значения возвращённого вызванным методом sum	∅

3.10 Алгоритм метода sum класса Object

Функционал: вычислить сумму элементов массива array.

Параметры: .

Возвращаемое значение: int (целочисленное значение суммы элементов массива).

Алгоритм метода представлен в таблице 10.

Таблица 10 – Алгоритм метода sum класса Object

№	**Предикат**	**Действия**	**№ перехода**
1		Инициализация целочисленной переменной sum значением 0	2
2		Инициализация целочисленной переменной i значением 0	3
3	Значение переменной i меньше значения поля size	Присваивание переменной sum суммы значения переменной sum и значения i-того элемента массива array	4
			5
4		Увеличение значения переменной i на 1	3
5		Возврат значения переменной sum	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-12.

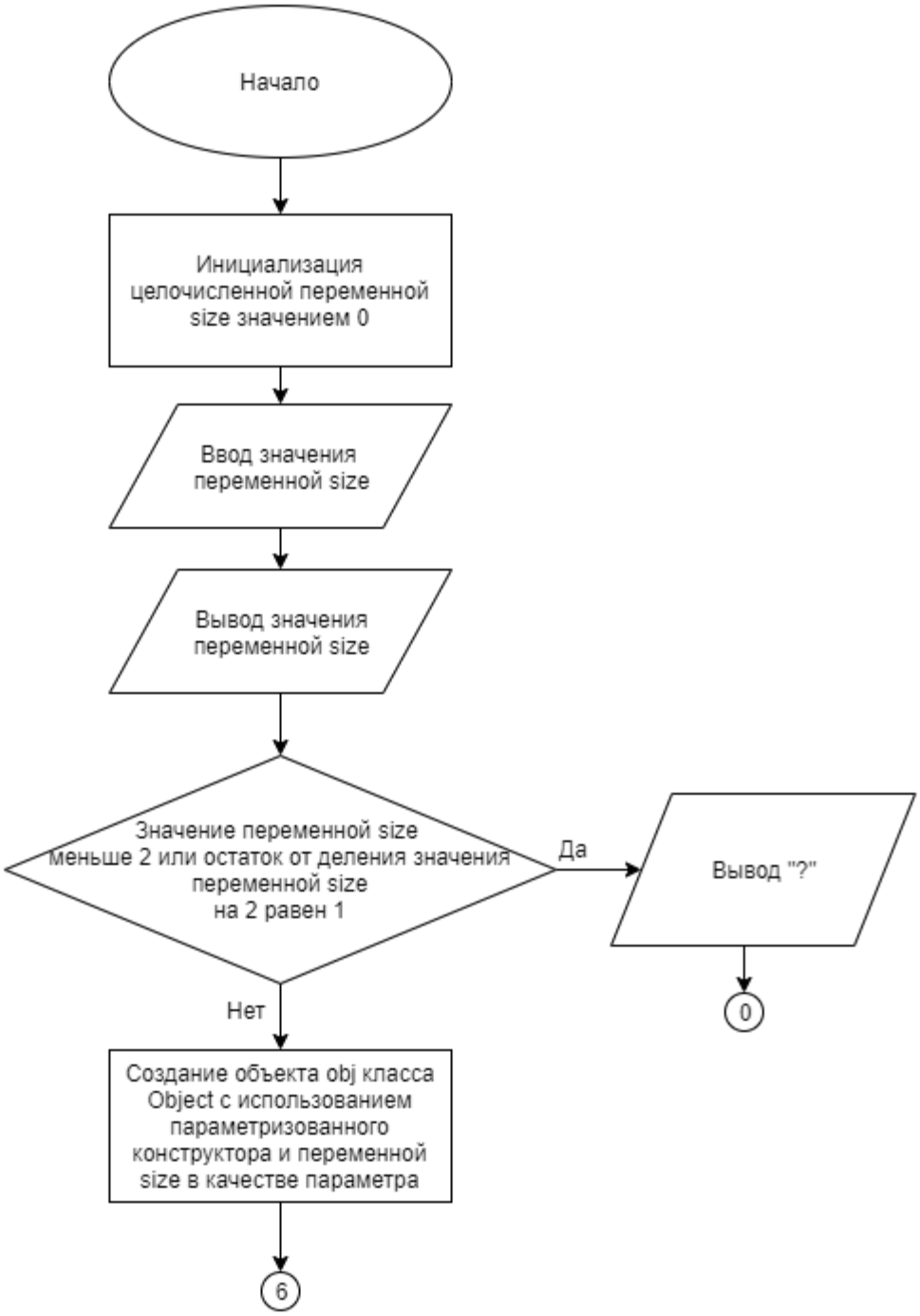

Рисунок 1 – Блок-схема алгоритма

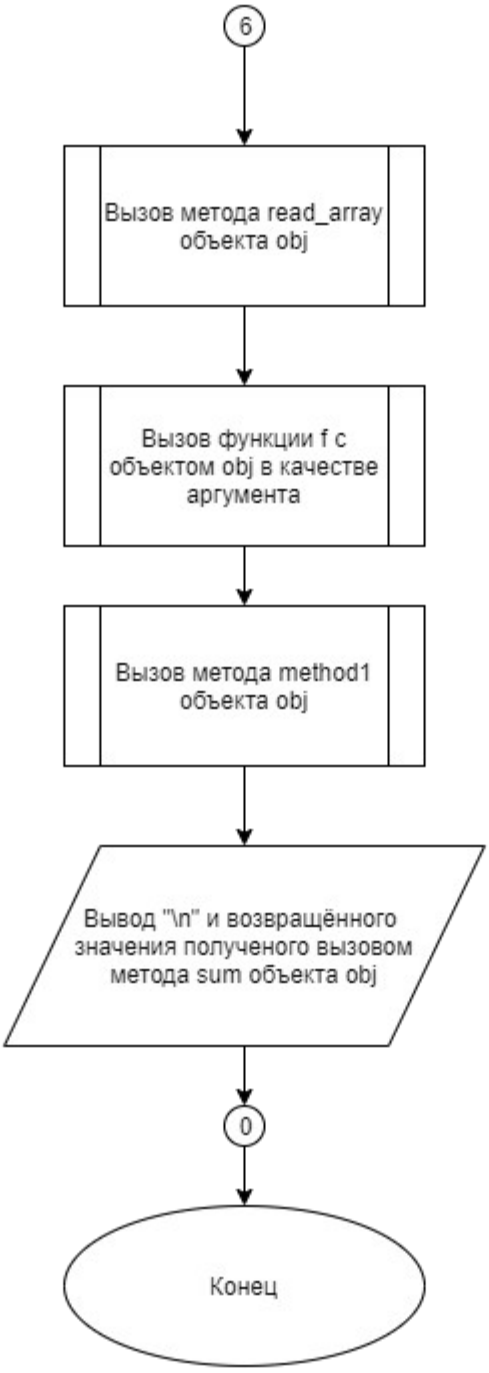

Рисунок 2 – Блок-схема алгоритма

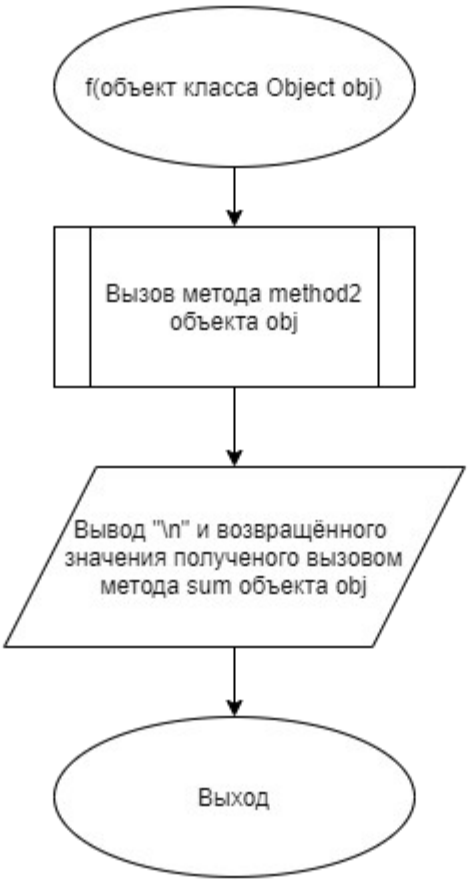

Рисунок 3 – Блок-схема алгоритма

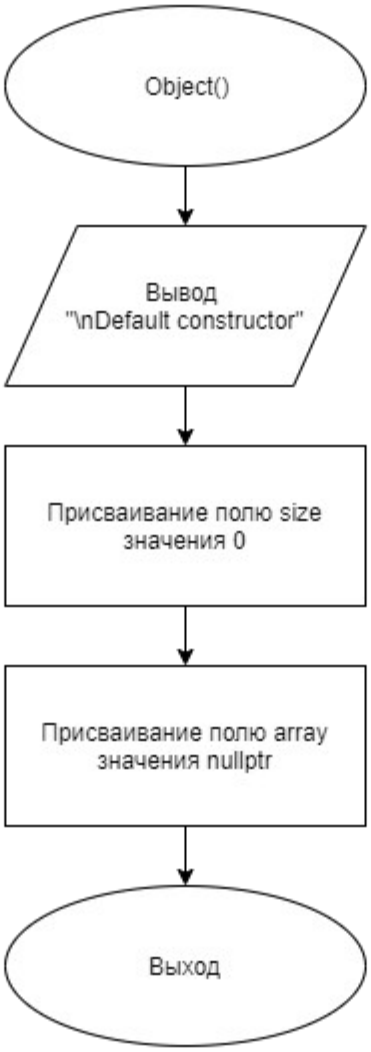

Рисунок 4 – Блок-схема алгоритма

Рисунок 5 – Блок-схема алгоритма

Рисунок 6 – Блок-схема алгоритма

Рисунок 7 – Блок-схема алгоритма

Рисунок 8 – Блок-схема алгоритма

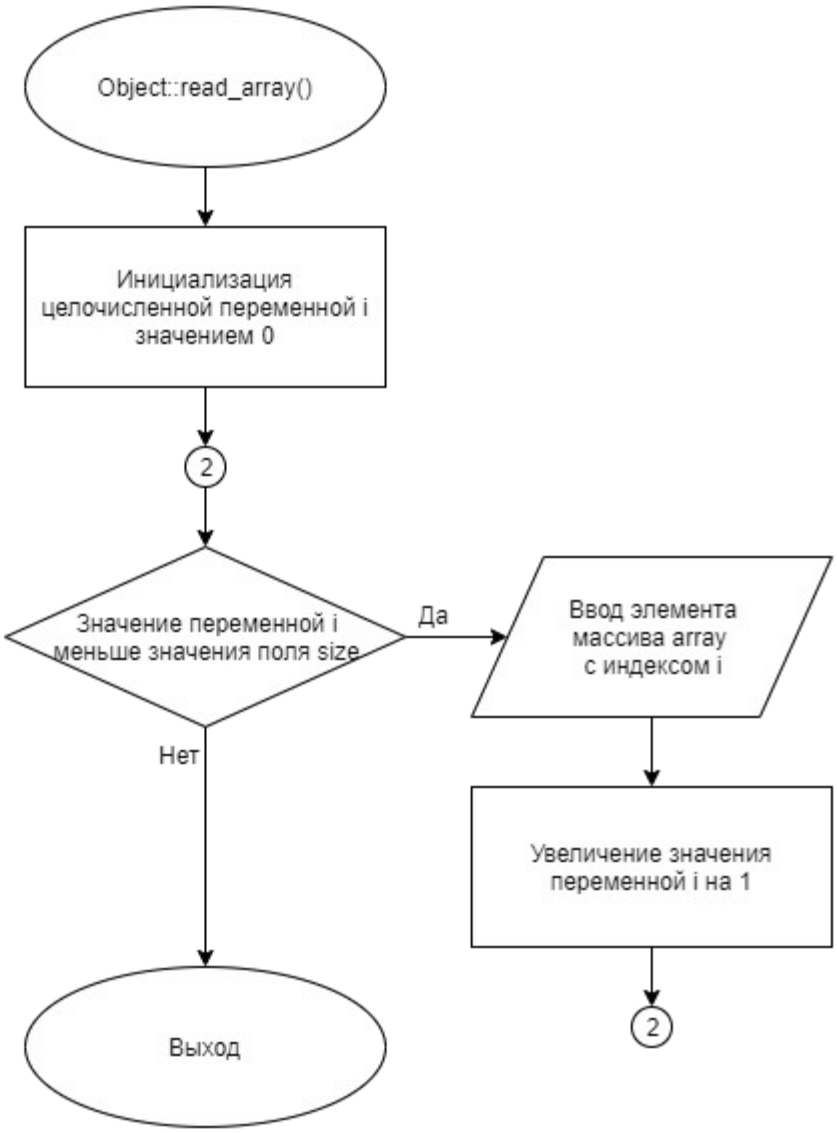

Рисунок 9 – Блок-схема алгоритма

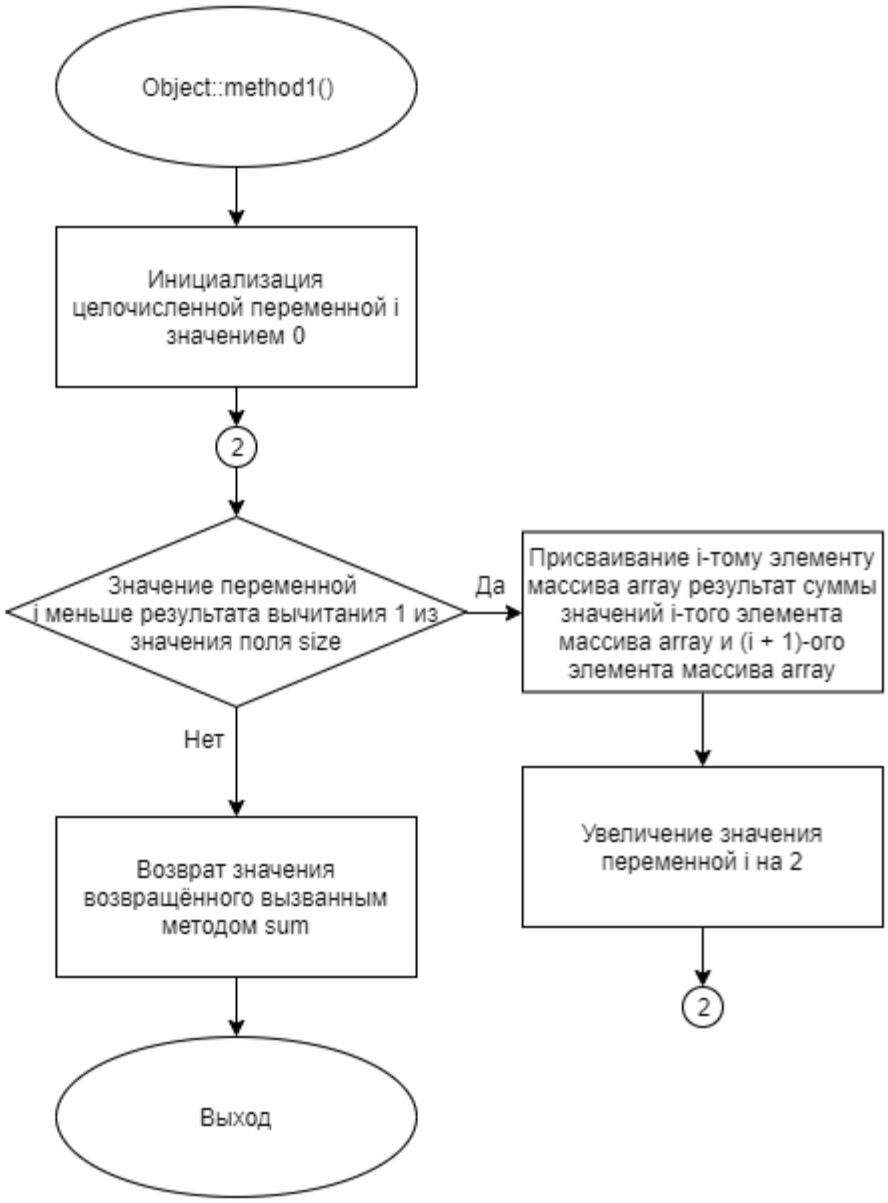

Рисунок 10 – Блок-схема алгоритма

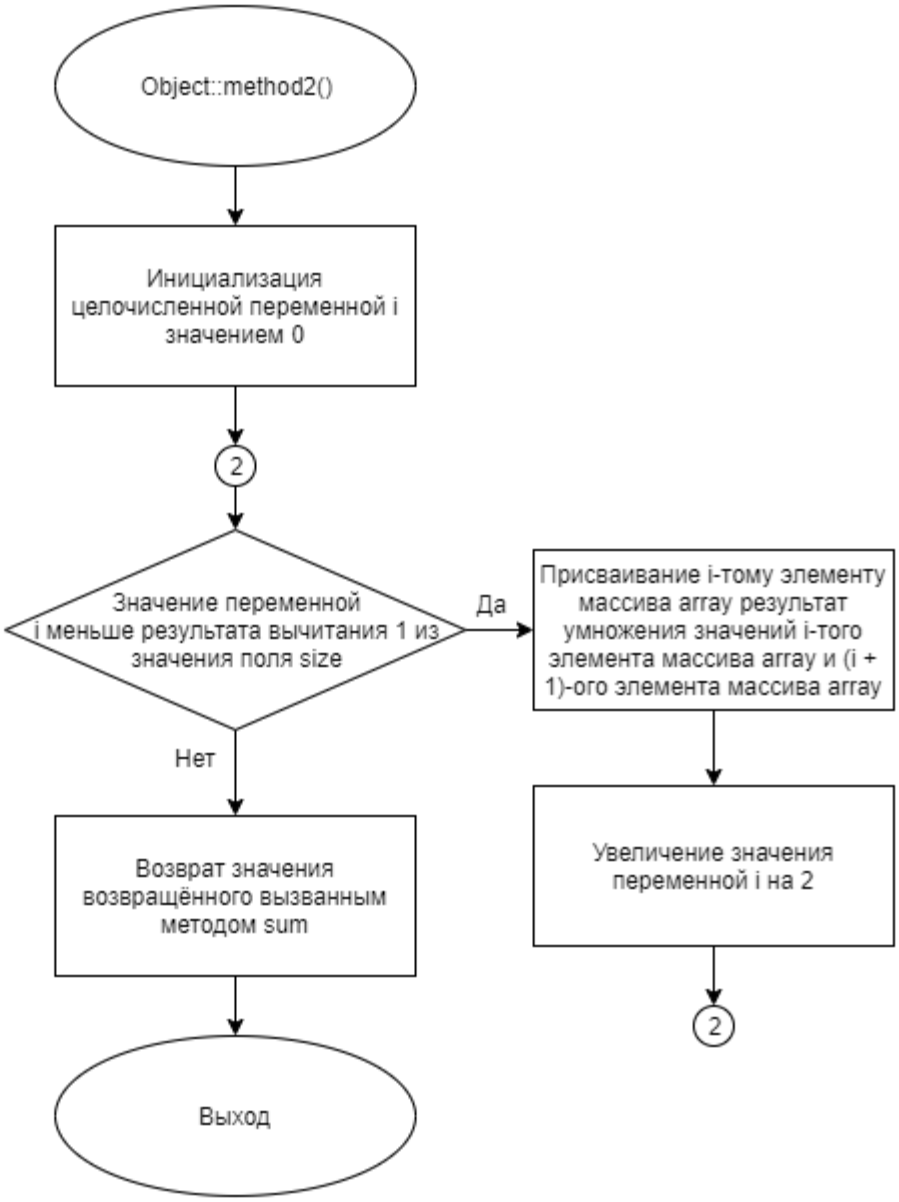

Рисунок 11 – Блок-схема алгоритма

Рисунок 12 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include <iostream>
#include "Object.h"

void f(Object obj)
{
	// Вызов метода method2
	// объекта obj
	obj.method2();
	// Переход на новую строку
	// И вывод значения возвращённого
	// методом sum объекта obj
	std::cout << '\n' << obj.sum();
}

int main()
{
	// Инициализация целочисленной переменной
	// size значением 0
	int size = 0;
	// Ввод size
	std::cin >> size;
	// Вывод size
	std::cout << size;

	// Если значение size меньше
	// двух или нечётно
	if (size < 2 || size % 2 == 1)
	{
		// Вывод "?"
		std::cout << '?';
	}
	// иначе
	else
	{
		// Создание объекта obj
		// класса Object
		Object obj(size);
		// Вызов метода
		// read_array объекта obj
		obj.read_array();
```

```
            // Вызов функции f
            // с объектом obj
            // в качестве аргумента
            f(obj);

            // Вызов метода
            // method1 объекта obj
            obj.method1();
            // Переход на новую строку
            // И вывод значения возвращённого
            // методом sum объекта obj
            std::cout << '\n' << obj.sum();
      }
      return 0;
}
```

5.2 Файл Object.cpp

Листинг 2 – Object.cpp

```
#include "Object.h"

Object::Object()
{
      std::cout << "\nDefault constructor";
      size = 0;
      array = nullptr;
}

Object::Object(int size)
{
      std::cout << "\nConstructor set";
      this->size = size;
      array = new int[size];
}

Object::Object(const Object& obj)
{
      std::cout << "\nCopy constructor";
      size = obj.size;
      array = new int[size];
      for (int i = 0; i < size; i++)
      {
            array[i] = obj.array[i];
      }
}

Object::~Object()
{
      std::cout << "\nDestructor";
      delete[] array;
}
```

```
void Object::read_array()
{
	for (int i = 0; i < size; i++)
	{
		std::cin >> array[i];
	}
}

int Object::method1()
{
	for (int i = 0; i < size - 1; i += 2)
	{
		array[i] = array[i] + array[i + 1];
	}

	return sum();
}

int Object::method2()
{
	for (int i = 0; i < size - 1; i += 2)
	{
		array[i] = array[i] * array[i + 1];
	}

	return sum();
}

int Object::sum()
{
	int sum = 0;
	for (int i = 0; i < size; i++)
	{
		sum += array[i];
	}

	return sum;
}
```

5.3 Файл Object.h

Листинг 3 – Object.h

```
#ifndef OBJECT_H
#define OBJECT_H

#include <iostream>

class Object
{
private:
	int* array;
	int size;
```

```
public:
      Object();
      Object(int size);
      Object(const Object& obj);
      ~Object();
      void read_array();
      int method1();
      int method2();
      int sum();
};

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 11.

Таблица 11 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
8 1 2 3 4 5 6 7 8	8 Constructor set Copy constructor 120 Destructor 56 Destructor	8 Constructor set Copy constructor 120 Destructor 56 Destructor
0	0?	0?
3	3?	3?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.

2. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.

3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avrora.ru/student/files/methodichescoe_posobie_dlya_laboratorny h_rabot_3.pdf (дата обращения 05.05.2021).

4. 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).

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).