Здесь будет титульник, листай ниже

**СОДЕРЖАНИЕ**

[1 ПОСТАНОВКА ЗАДАЧИ 5](#__RefHeading___Toc3862_1011462937)

[1.1 Описание входных данных 6](#__RefHeading___Toc3864_1011462937)

[1.2 Описание выходных данных 6](#__RefHeading___Toc3866_1011462937)

[2 МЕТОД РЕШЕНИЯ 8](#__RefHeading___Toc3868_1011462937)

[3 ОПИСАНИЕ АЛГОРИТМОВ 10](#__RefHeading___Toc3870_1011462937)

[3.1 Алгоритм функции main 10](#__RefHeading___Toc3872_1011462937)

[3.2 Алгоритм функции Func 11](#__RefHeading___Toc3874_1011462937)

[3.3 Алгоритм конструктора класса Object 11](#__RefHeading___Toc3876_1011462937)

[3.4 Алгоритм деструктора класса Object 12](#__RefHeading___Toc3878_1011462937)

[3.5 Алгоритм метода FillArray класса Object 13](#__RefHeading___Toc3880_1011462937)

[3.6 Алгоритм метода Sum класса Object 13](#__RefHeading___Toc3882_1011462937)

[3.7 Алгоритм метода Method1 класса Object 14](#__RefHeading___Toc3884_1011462937)

[3.8 Алгоритм метода Method2 класса Object 15](#__RefHeading___Toc3886_1011462937)

[3.9 Алгоритм конструктора класса Object 15](#__RefHeading___Toc3888_1011462937)

[3.10 Алгоритм конструктора класса Object 16](#__RefHeading___Toc3890_1011462937)

[4 БЛОК-СХЕМЫ АЛГОРИТМОВ 17](#__RefHeading___Toc3892_1011462937)

[5 КОД ПРОГРАММЫ 23](#__RefHeading___Toc3894_1011462937)

[5.1 Файл main.cpp 23](#__RefHeading___Toc3896_1011462937)

[5.2 Файл Object.cpp 23](#__RefHeading___Toc3898_1011462937)

[5.3 Файл Object.h 25](#__RefHeading___Toc3900_1011462937)

[6 ТЕСТИРОВАНИЕ 26](#__RefHeading___Toc3902_1011462937)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 27](#__RefHeading___Toc3904_1011462937)

# 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.
* 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

Для решения задачи используется библиотека iostream, пространство имен std, оператор цикла со счётчиком for, условная конструкция if\_else, объекты стандартных потоков ввода/вывода cin/cout.

Класс Object:

* Свойства/поля:
  + Поле, отвечающее за объявление указателя на динамический массив:
    - Наименование - arr;
    - Тип - int\*;
    - Модификатор доступа - private;
  + Поле, отвечающее за хранения размера массива arr - приватного поля класса Object:
    - Наименование - arrSize;
    - Тип - int;
    - Модификатор доступа - private;
* Методы:
  + Конструктор Object
    - Функционал - Конструктор по умолчанию, выводящий сообщение на экран.
  + Конструктор Object
    - Функционал - Параметризированный конструктор с целочисленным параметром, по значению которого определяется размер целочисленного массива arr из закрытой области.
  + Конструктор Object
    - Функционал - Конструктор копирования объекта класса Object для корректного выполнения вычислений в методах класса.
  + Деструктор ~Object
    - Функционал - Деструктор, выводящий сообщение на экран и очищающий память, выделенную под динамический массив arr.
  + Метод FillArray
    - Функционал - Используется для заполнения массива arr значениями, введенными с клавиатуры пользователем.
  + Метод Sum
    - Функционал - Используется для суммирования значений элементов массива arr и возвращение результата суммирования.
  + Метод Method1
    - Функционал - Используется для суммирования значений очередной пары элементов и присваиваения результата суммы первому элементу пары.
  + Метод Method2
    - Функционал - Используется для умножения значений очередной пары элементов и присваиваения результата суммы первому элементу пары.

# 3 ОПИСАНИЕ АЛГОРИТМОВ

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

## Алгоритм функции main

Функционал: Основной алгоритм программы.

Параметры: Отсутствуют.

Возвращаемое значение: int - индикатор корректности завершения программы.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Объявление целочисленной переменной size | 2 |
| 2 |  | Ввод значения переменной size с клавиатуры | 3 |
| 3 | Значение переменной size больше 2 и четное | Вывод значения переменной size на экран | 4 |
|  | Вывод значения переменной size и "?" на экран | 10 |
| 4 |  | Создание объекта obj класса Object, путём вызова параметризированного конструктора с аргументом в виде значения переменной size | 5 |
| 5 |  | Вызов метода FillArray объекта obj | 6 |
| 6 |  | Вызов функции Func, с передачей в качестве аргумента объект obj | 7 |
| 7 |  | Переход на новую строку | 8 |
| 8 |  | Вызов метода Method1 объекта obj | 9 |
| 9 |  | Вывод результата выполнения метода Method1 объекта obj на экран | 10 |
| 10 |  | Возвращение значения работы функции 0 | ∅ |

## Алгоритм функции Func

Функционал: Создание нового объекта через конструктор копирования и вызов методов данного объекта.

Параметры: Object obj - копия объекта класса Object, созданая через конструктор копирования.

Возвращаемое значение: Отсутствует.

Алгоритм функции представлен в таблице 2.

Таблица 2 – Алгоритм функции Func

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Вызов метода Method2 объекта obj | 2 |
| 2 |  | Вывод результата выполнения метода Method2 объекта obj на экран | ∅ |

## Алгоритм конструктора класса Object

Функционал: Конструктор для копирования значений полей передаваемого объекта создаваемому объекту.

Параметры: const Object &obj - объект, значения полей которого используются при создании копии данного объекта.

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса Object

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Вывод сообщения "Copy constructor" на экран | 2 |
| 2 |  | Присваивание полю arrSize создаваемого объекта значение поля arrSize объекта копирования | 3 |
| 3 |  | Выделение памяти под динамический массив arr создаваемого объекта с размером arrSize через оператор new | 4 |
| 4 |  | Инициализация целочисленной переменной i значением 0 | 5 |
| 5 | Значение переменной i меньше arrSize | Присваивание i-той ячейке массива arr создаваемого объекта значение i-той ячейки массива arr объекта  копирования obj | 6 |
|  |  | ∅ |
| 6 |  | Инкремент значения переменной i | 5 |

## Алгоритм деструктора класса Object

Функционал: Удаление объекта и его полей из памяти.

Параметры: Отсутствуют.

Алгоритм деструктора представлен в таблице 4.

Таблица 4 – Алгоритм деструктора класса Object

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Вывод сообщения "Destructor" на экран | 2 |
| 2 | Значение поля arr объекта не равно значению nullptr | Очищение выделенной под поле arr памяти с помощью оператора delete | ∅ |
|  |  | ∅ |

## Алгоритм метода FillArray класса Object

Функционал: Заполнение массива arr значениями, введенными с клавиатуры пользователем.

Параметры: Отсутствуют.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода FillArray класса Object

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Инициализация целочисленной переменной i значением 0 | 2 |
| 2 | Значение переменной i меньше arrSize | Ввод значения i-той ячейки массива arr | 3 |
|  |  | ∅ |
| 3 |  | Инкремент значения переменной i | 2 |

## Алгоритм метода Sum класса Object

Функционал: Cуммирование значений элементов массива arr и возвращение результата суммирования.

Параметры: Отсутствуют.

Возвращаемое значение: int - результат суммирования элементов массива arr.

Алгоритм метода представлен в таблице 6.

Таблица 6 – 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 arrSize | Присваивание переменной sum значение суммы sum и i-того элемента массива arr | 4 |
|  |  | 5 |
| 4 |  | Инкремент значения переменной i | 3 |
| 5 |  | Возвращение значения переменной sum | ∅ |

## Алгоритм метода Method1 класса Object

Функционал: Суммирование значений очередной пары элементов и присваиваение результата суммы первому элементу пары.

Параметры: Отсутствуют.

Возвращаемое значение: int - результат суммирования элементов массива arr.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода Method1 класса Object

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Инициализация целочисленной переменной i значением 0 | 2 |
| 2 | Значение переменной i меньше arrSize | Присваивание i-тому элементу массива arr значение суммы текущего и следующего элементов | 3 |
|  |  | 4 |
| 3 |  | Увеличение значения переменной i на 2 | 2 |
| 4 |  | Вызов метода Sum и возвращение результата его выполнения | ∅ |

## Алгоритм метода Method2 класса Object

Функционал: Умножение значений очередной пары элементов и присваиваение результата произведения первому элементу пары.

Параметры: Отсутствуют.

Возвращаемое значение: int - результат суммирования элементов массива arr.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода Method2 класса Object

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Инициализация целочисленной переменной i значением 0 | 2 |
| 2 | Значение переменной i меньше arrSize | Присваивание i-тому элементу массива arr значение произведения текущего и следующего элементов | 3 |
|  |  | 4 |
| 3 |  | Увеличение значения переменной i на 2 | 2 |
| 4 |  | Вызов метода Sum и возвращение результата его выполнения | ∅ |

## Алгоритм конструктора класса Object

Функционал: Создание объекта на основе класса Object..

Параметры: int arrSize - размер поля arr класса Object, где arr - динамический целочисленный массив.

Алгоритм конструктора представлен в таблице 9.

Таблица 9 – Алгоритм конструктора класса Object

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Вывод сообщения "Constructor set" на экран | 2 |
| 2 |  | Выделение памяти под динамический массив arr размера arrSize через оператор new | 3 |
| 3 |  | Присваивание полю arr класса значение параметра arrSize | ∅ |

## Алгоритм конструктора класса Object

Функционал: Вывод информации о работе конструктора по умолчанию.

Параметры: Отсутствуют.

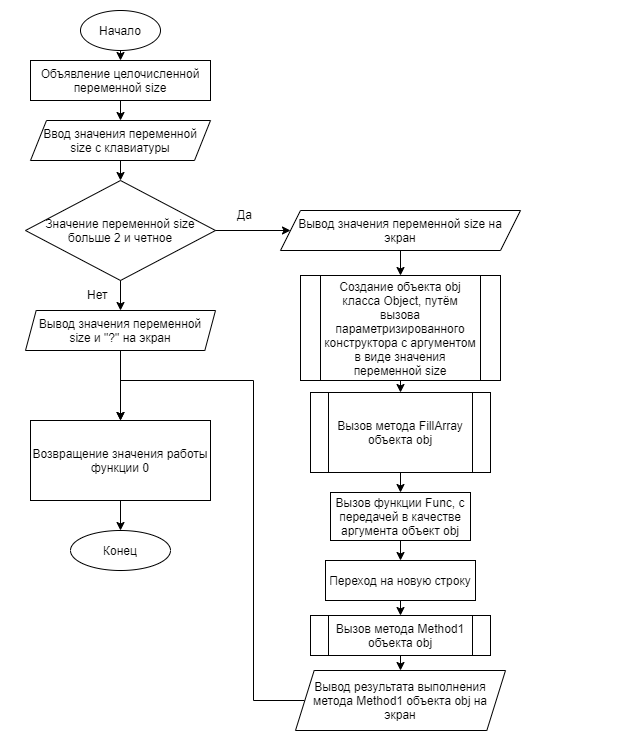
Алгоритм конструктора представлен в таблице 10.

Таблица 10 – Алгоритм конструктора класса Object

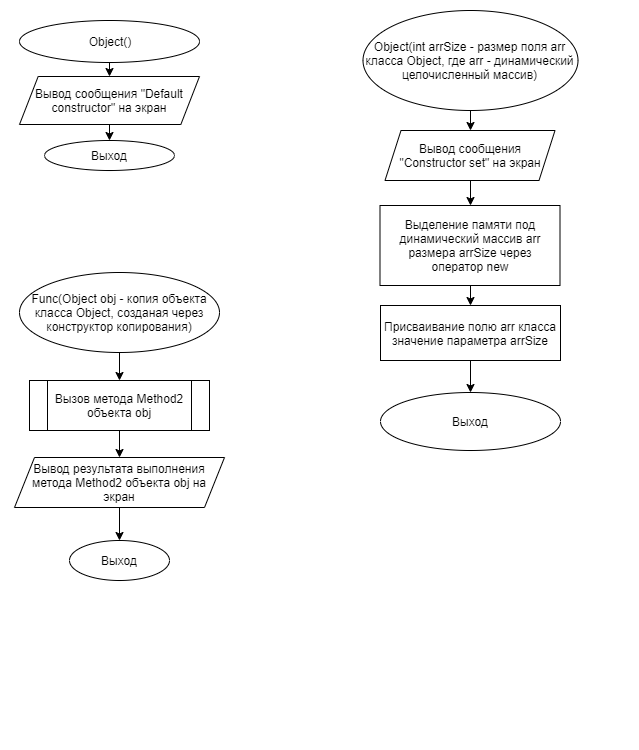
| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Вывод сообщения "Default constructor" на экран | ∅ |

# 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

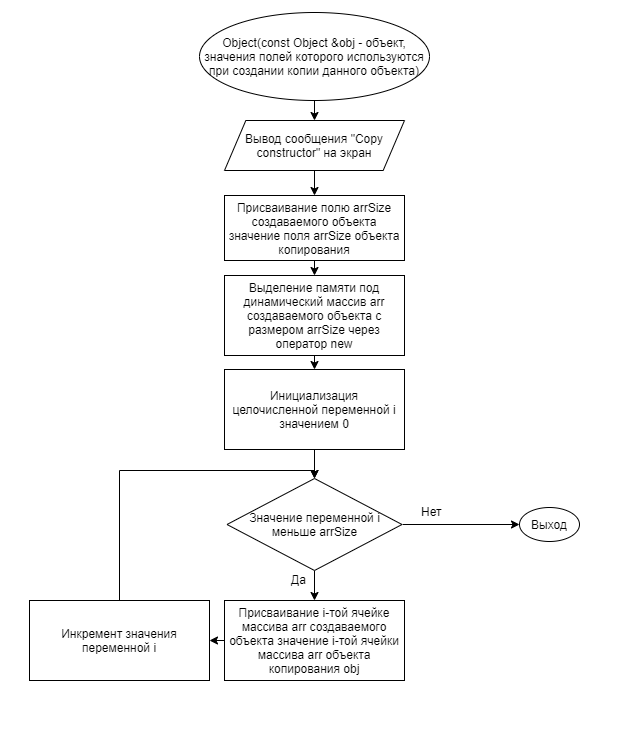
Представим описание алгоритмов в графическом виде на рисунках 1-6.



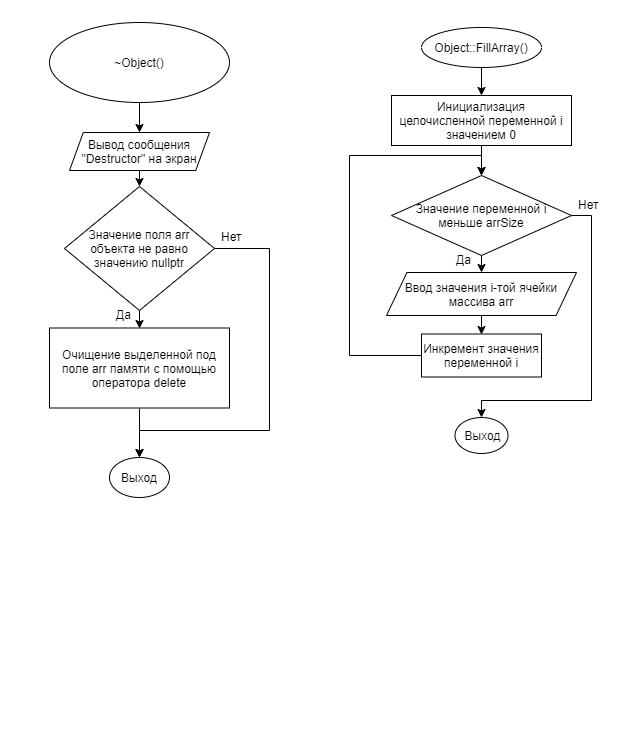
**Рисунок 1 – Блок-схема алгоритма**



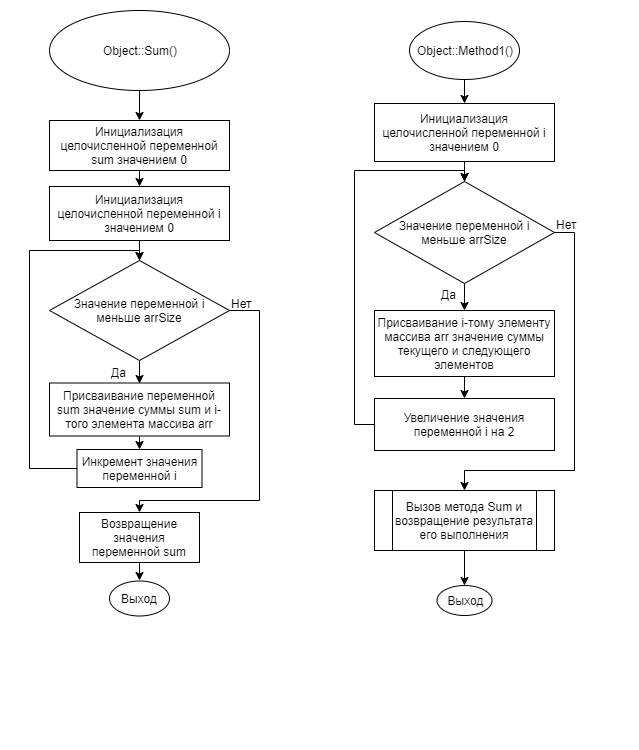
**Рисунок 2 – Блок-схема алгоритма**



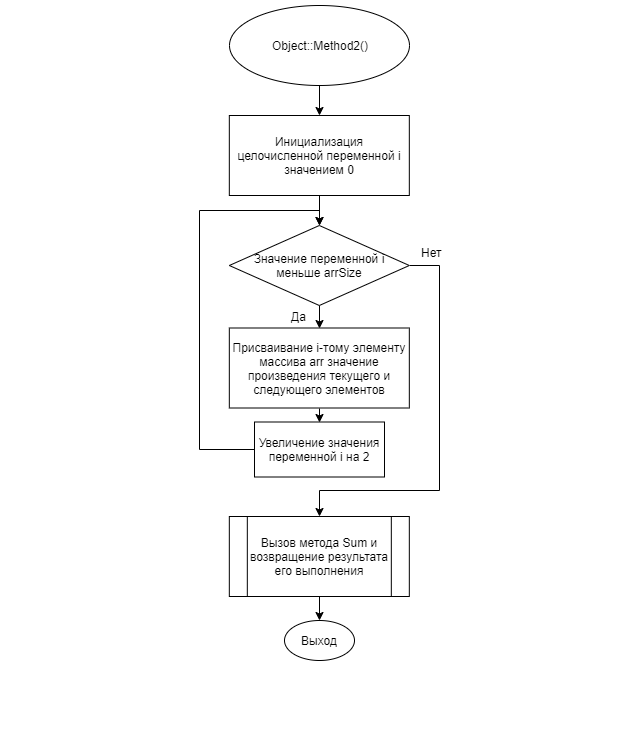
**Рисунок 3 – Блок-схема алгоритма**



**Рисунок 4 – Блок-схема алгоритма**



**Рисунок 5 – Блок-схема алгоритма**



**Рисунок 6 – Блок-схема алгоритма**

# 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

## **Файл** **main.cpp**

Листинг 1 – main.cpp

|  |
| --- |
| #include <iostream>  #include "Object.h"  using namespace std;  void Func(Object obj)  {  cout << obj.Method2() << endl;  }  int main()  {  int size;  cin >> size;  if(size > 2 && size%2 == 0)  {  cout << size << endl;  Object obj(size);  obj.FillArray();  Func(obj);  cout << endl;  cout << obj.Method1() << endl;  }  else  {  cout << size << "?";  }  return(0);  } |

## **Файл** **Object.cpp**

Листинг 2 – Object.cpp

|  |
| --- |
| #include "Object.h"  #include <iostream>  using namespace std;  Object::Object()  {  cout << "Default constructor" << endl;  }  Object::Object(int arrSize)  {  cout << "Constructor set" << endl;  arr = new int[arrSize];  this->arrSize = arrSize;  }  Object::Object(const Object &obj)  {  cout << "Copy constructor" << endl;  arrSize = obj.arrSize;  arr = new int[arrSize];  for(int i = 0; i < arrSize; i++)  {  arr[i] = obj.arr[i];  }  }  Object::~Object()  {  cout << "Destructor";  if (arr != nullptr)  delete[] arr;  }  void Object::FillArray()  {  for(int i = 0; i < arrSize; i++)  {  cin >> arr[i];  }  }  int Object::Sum()  {  int sum = 0;  for(int i = 0; i < arrSize; i++)  {  sum += arr[i];  }  return sum;  }  int Object::Method1()  {  for(int i = 0; i < arrSize; i+=2)  {  arr[i] += arr[i+1];  }  return Sum();  }  int Object::Method2()  {  for(int i = 0; i < arrSize; i+=2)  {  arr[i] \*= arr[i+1];  }  return Sum();  } |

## **Файл** **Object.h**

Листинг 3 – Object.h

|  |
| --- |
| #ifndef OBJECT\_H  #define OBJECT\_H  class Object  {  int\* arr;  int arrSize;  public:  Object();  Object(int arrSize);  Object(const Object &obj);  ~Object();  void FillArray();  int Sum();  int Method1();  int Method2();  };  #endif |

# 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 11.

Таблица 11 – Результат тестирования программы

| Входные данные | Ожидаемые выходные данные | Фактические выходные данные |
| --- | --- | --- |
| 1 | 1? | 1? |
| 2 | 2? | 2? |
| 3 | 3? | 3? |
| 4  1 2 3 4 | 4  Constructor set  Copy constructor  20  Destructor  16  Destructor | 4  Constructor set  Copy constructor  20  Destructor  16  Destructor |
| 5 | 5? | 5? |
| 6  1 2 3 4 5 6 | 6  Constructor set  Copy constructor  56  Destructor  33  Destructor | 6  Constructor set  Copy constructor  56  Destructor  33  Destructor |
| 7 | 7? | 7? |
| 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 |
| 9 | 9? | 9? |
| 10  1 2 3 4 5 6 7 8 9 10 | 10  Constructor set  Copy constructor  220  Destructor  85  Destructor | 10  Constructor set  Copy constructor  220  Destructor  85  Destructor |

# 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. С++: 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).