Здесь будет титульник, листай ниже

**СОДЕРЖАНИЕ**

[1 ПОСТАНОВКА ЗАДАЧИ 5](#__RefHeading___Toc2605_2581197707)

[1.1 Описание входных данных 6](#__RefHeading___Toc2607_2581197707)

[1.2 Описание выходных данных 7](#__RefHeading___Toc2609_2581197707)

[2 МЕТОД РЕШЕНИЯ 9](#__RefHeading___Toc2611_2581197707)

[3 ОПИСАНИЕ АЛГОРИТМОВ 10](#__RefHeading___Toc2613_2581197707)

[3.1 Алгоритм функции main 10](#__RefHeading___Toc2615_2581197707)

[3.2 Алгоритм функции Func 11](#__RefHeading___Toc2617_2581197707)

[3.3 Алгоритм конструктора класса Object 12](#__RefHeading___Toc2619_2581197707)

[3.4 Алгоритм метода CreateArray класса Object 12](#__RefHeading___Toc2621_2581197707)

[3.5 Алгоритм метода Object класса OutputArray 12](#__RefHeading___Toc2623_2581197707)

[4 БЛОК-СХЕМЫ АЛГОРИТМОВ 14](#__RefHeading___Toc2625_2581197707)

[5 КОД ПРОГРАММЫ 17](#__RefHeading___Toc2627_2581197707)

[5.1 Файл main.cpp 17](#__RefHeading___Toc2629_2581197707)

[5.2 Файл Object.cpp 18](#__RefHeading___Toc2631_2581197707)

[5.3 Файл Object.h 19](#__RefHeading___Toc2633_2581197707)

[6 ТЕСТИРОВАНИЕ 20](#__RefHeading___Toc2635_2581197707)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 21](#__RefHeading___Toc2637_2581197707)

# 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, вначале работы выдает сообщение;
* Параметризированный конструктор, передается целочисленный параметр. Параметр должен иметь значение больше 2 и быть четным. Вначале работы выдает сообщение;
* Конструктор копии, обеспечивает создание копии объекта в новой области памяти. Вначале работы выдает сообщение;
* Метод деструктор, который в начале работы выдает сообщение;
* Метод который создает целочисленный массив в закрытой области, согласно ранее заданной размерности.
* Метод ввода данных для созданного массива;
* Метод 1, который суммирует значения очередной пары элементов и сумму присваивает первому элементу пары. 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. 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;
* Метод последовательного вывода содержимого элементов массива, которые разделены тремя пробелами.

Разработать функцию func, которая имеет один целочисленный параметр, содержащий размерность массива. В функции должен быть реализован алгоритм:

* Создание локального объекта с использованием параметризированного конструктора.
* Возврат созданного локального объекта.

В основной функции реализовать алгоритм:

* Ввод размерности массива.
* Если размерность массива некорректная, вывод сообщения и завершить работу алгоритма.
* Вывод значения размерности массива.
* Создание первого объекта.
* Присвоение первому объекту результата работы функции func с аргументом, содержащим значение размерности массива.
* Для первого объекта вызов метода создания массива.
* Для первого объекта вызов метода ввода данных массива.
* Для первого объекта вызов метода 2.
* Инициализация второго объекта первым объектом.
* Вызов метода 1 для второго объекта.
* Вывод содержимого массива первого объекта.
* Вывод суммы элементов массива первого объекта.
* Вывод содержимого массива второго объекта.
* Вывод суммы элементов массива второго объекта.

## 1.1 Описание входных данных

Первая строка:

«Целое число»

Вторая строка:

«Целое число» «Целое число» . . .

**Пример:**

4

3 5 1 2

## 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 копии в начале работы с новой строки выдает сообщение:

Copy constructor

Деструктор в начале работы с новой строки выдает сообщение:

Destructor

Метод последовательного вывода содержимого элементов массива, с новой строки выдает:

«Целое число» «Целое число» «Целое число» . . .

**Пример вывода:**

4

Default constructor

Constructor set

Destructor

Copy constructor

15 5 2 2

24

20 5 4 2

31

Destructor

Destructor

# 2 МЕТОД РЕШЕНИЯ

Для решения задачи используется то же, что и в предыдущей задаче 3\_1\_3, однако класс Object, функции main и Func переработаны под условия задачи.

Класс Object:

* Свойства/поля:
  + Поля arr и arrSize класса Object остались неизменными.
* Методы:
  + Те же методы, что и в предыдущей задаче 3\_1\_3.
  + Метод CreateArray
    - Функционал - Используется для создания целочисленного массива в закрытой области.
  + Метод OutputArray
    - Функционал - Используется для последовательного вывода содержимого элементов массива, которые разделены тремя пробелами

# 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 работы 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 |  | Начало работы алгоритма из предыдущей задачи 3\_1\_3 | 2 |
| 2 |  | Создание объекта obj1 класса Object, путём вызова конструктора по умолчанию | 3 |
| 3 |  | Присваивание объекту obj1 значение выполнения функции Func с аргументом в виде значения переменной size | 4 |
| 4 |  | Вызов метода CreateArray объекта obj1 | 5 |
| 5 |  | Вызов метода FillArray объекта obj1 | 6 |
| 6 |  | Вызов метода Method2 объекта obj1 | 7 |
| 7 |  | Создание объекта obj2 класса Object, путём вызова конструктора копирования с аргументом в виде объекта obj1 | 8 |
| 8 |  | Вызов метода Method1 объекта obj2 | 9 |
| 9 |  | Вызов метода OutputArray объекта obj1 | 10 |
| 10 |  | Вызов метода Sum объекта obj1 | 11 |
| 11 |  | Вывод результата выполнения метода Sum объекта obj1 на экран | 12 |
| 12 |  | Вызов метода OutputArray объекта obj2 | 13 |
| 13 |  | Вызов метода Sum объекта obj2 | 14 |
| 14 |  | Вывод результата выполнения метода Sum объекта obj2 на экран | 15 |
| 15 |  | Окончание работы алгоритма из предыдущей задачи 3\_1\_3 | ∅ |

## Алгоритм функции Func

Функционал: Создание локального объекта с использованием параметризированного конструктора.

Параметры: int size - параметр для задания размера массива создаваемого объекта.

Возвращаемое значение: Object - возврат значения созданного объекта класса Object.

Алгоритм функции представлен в таблице 2.

Таблица 2 – Алгоритм функции Func

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Создание объекта newObj класса Object, путём вызова параметризированного конструктора с аргументом в виде параметра  size | 2 |
| 2 |  | Возврат значения созданного объекта newObj класса Object | ∅ |

## 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 3.

Таблица 3 – Алгоритм конструктора класса Object

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Вывод сообщения "Constructor set" на экран | 2 |
| 2 |  | Присваивание полю arrSize текущего класса значение параметра arrSize | ∅ |

## Алгоритм метода CreateArray класса Object

Функционал: Создание целочисленного массива в закрытой области.

Параметры: Отсутствуют.

Возвращаемое значение: Отсутствует.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода CreateArray класса Object

| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Выделение памяти под динамический массив arr размера arrSize через оператор new | ∅ |

## Алгоритм метода Object класса OutputArray

Функционал: Последовательный вывод содержимого элементов массива, которые разделены тремя пробелами.

Параметры: Отсутствуют.

Возвращаемое значение: Отсутствует.

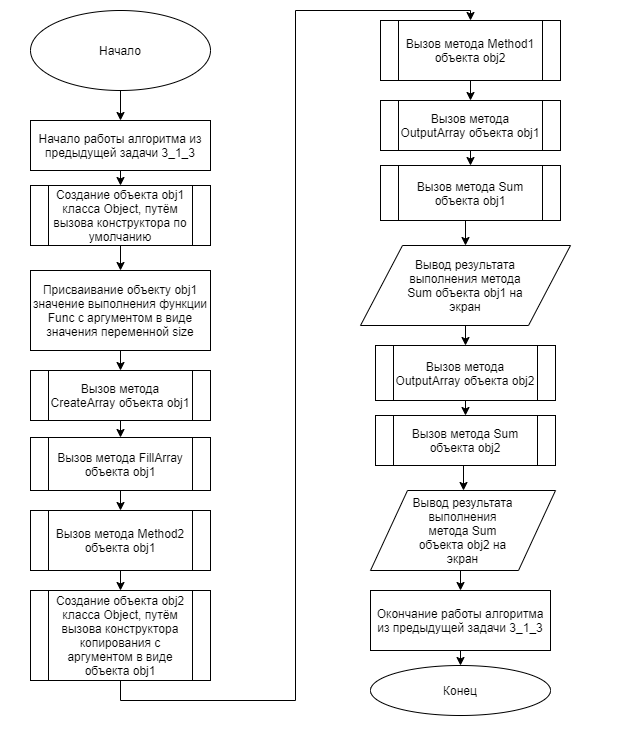
Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода Object класса OutputArray

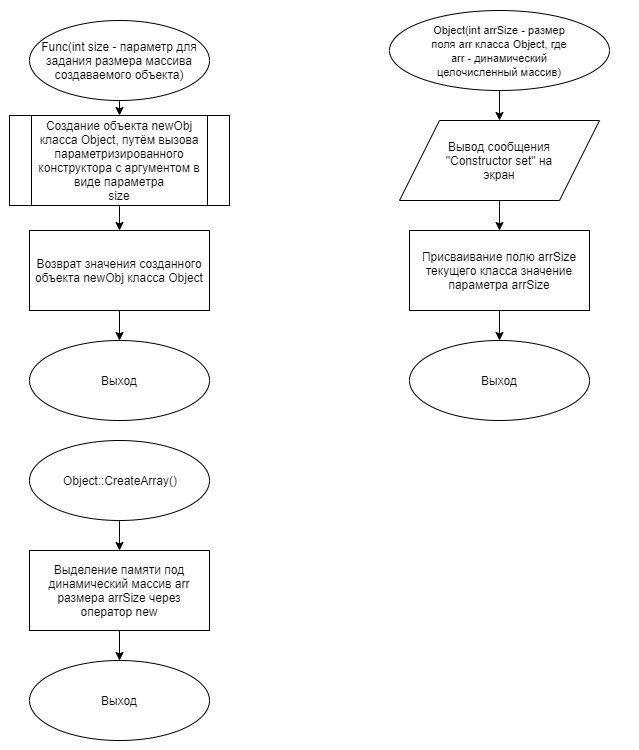
| № | Предикат | Действия | № перехода |
| --- | --- | --- | --- |
| 1 |  | Вывод первого элемента массива arr на экран | 2 |
| 2 |  | Инициализация целочисленной переменной i значением 1 | 3 |
| 3 | Значение переменной i меньше значения поля arrSize | Вывод трёх пробелов и i-того элемента массива arr | 4 |
|  | Переход на новую строку | ∅ |
| 4 |  | Инкремент значения переменной i | 3 |

# 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

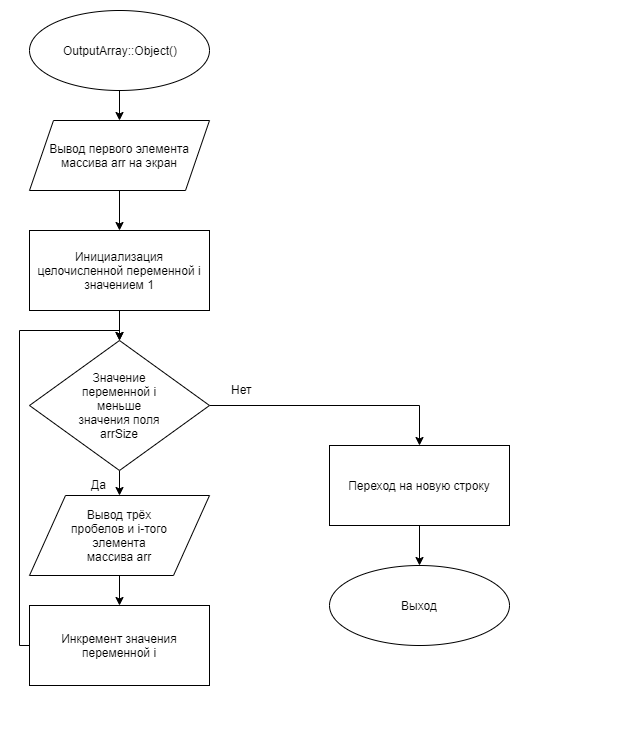
Представим описание алгоритмов в графическом виде на рисунках 1-3.



**Рисунок 1 – Блок-схема алгоритма**



**Рисунок 2 – Блок-схема алгоритма**



**Рисунок 3 – 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;  Object Func(int size)  {  Object newObj(size);  return newObj;  }  int main()  {  int size;  cin >> size;  if(size > 2 && size%2 == 0)  {  cout << size << endl;  Object obj1;  obj1 = Func(size);  obj1.CreateArray();  obj1.FillArray();  obj1.Method2();  Object obj2(obj1);  obj2.Method1();  obj1.OutputArray();  cout << obj1.Sum() << endl;  obj2.OutputArray();  cout << obj2.Sum();  }  else  {  cout << size << "?";  }  return(0);  } |

## **Файл** **Object.cpp**

Листинг 2 – Object.cpp

|  |
| --- |
| #include "Object.h"  Object::Object()  {  cout << "Default constructor" << endl;  }  Object::Object(int arrSize)  {  cout << "Constructor set";  this->arrSize = arrSize;  }  Object::Object(const Object &obj)  {  cout << "\nCopy constructor" << endl;  arrSize = obj.arrSize;  arr = new int[arrSize];  for(int i = 0; i < arrSize; i++)  {  arr[i] = obj.arr[i];  }  }  Object::~Object()  {  cout << "\nDestructor";  }  void Object::FillArray()  {  for(int i = 0; i < arrSize; i++)  {  cin >> arr[i];  }  }  void Object::CreateArray()  {  arr = new int[arrSize];  }  void Object::OutputArray()  {  cout << arr[0];  for(int i = 1; i < arrSize; i++)  {  cout << " " << arr[i];  }  cout << endl;  }  int Object::Sum()  {  int sum = 0;  for(int i = 0; i < arrSize; i++)  {  sum += arr[i];  }  return sum;  }  void Object::Method1()  {  for(int i = 0; i < arrSize; i+=2)  {  arr[i] += arr[i+1];  }  }  void Object::Method2()  {  for(int i = 0; i < arrSize; i+=2)  {  arr[i] \*= arr[i+1];  }  } |

## **Файл** **Object.h**

Листинг 3 – Object.h

|  |
| --- |
| #ifndef OBJECT\_H  #define OBJECT\_H  #include <iostream>  using namespace std;  class Object  {  int\* arr;  int arrSize;  public:  Object();  Object(int arrSize);  Object(const Object &obj);  ~Object();  void FillArray();  void CreateArray();  void OutputArray();  int Sum();  void Method1();  void Method2();  };  #endif |

# 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 6.

Таблица 6 – Результат тестирования программы

| Входные данные | Ожидаемые выходные данные | Фактические выходные данные |
| --- | --- | --- |
| 4  3 5 1 2 | 4  Default constructor  Constructor set  Destructor  Copy constructor  15 5 2 2  24  20 5 4 2  31  Destructor  Destructor | 4  Default constructor  Constructor set  Destructor  Copy constructor  15 5 2 2  24  20 5 4 2  31  Destructor  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).