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, 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 значений элементов 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 двумя пробелами;

- метод, который возвращает значение указателя на массив из закрытой области;
- метод, который присваивает значение указателя массива из закрытой области.

Назовём класс описания данного объекта cl_obj (для примера, у вас он может называться иначе).

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

1. Инициализация указателя на объект класса cl_obj адресом объекта, созданного с использованием параметризированного конструктора.
2. С использованием указателя на объект класса cl_obj вызов метода создания массива.
3. С использованием указателя на объект класса cl_obj вызов метода ввода значений элементов массива.
4. С использованием указателя на объект класса cl_obj вызов метода 2.
5. Возврат указателя на объект класса cl_obj.

В основной функции реализовать алгоритм:

1. Ввод размерности массива.
2. Если размерность массива некорректная, вывод сообщения и завершить работу алгоритма.
3. Вывод значения размерности массива.
4. Объявить первый указатель на объект класса cl_obj.
5. Присвоение первому указателю результата работы функции func с аргументом, содержащим значение размерности массива.
6. С использованием первого указателя вызов метода 1.
7. Инициализация второго указателя на объект класса cl_obj адресом

объекта, созданного с использованием конструктора копии с аргументом первого объекта.

8. С использованием второго указателя вызов метода 2.
9. Вывод содержимого массива первого объекта.
10. Вывод суммы элементов массива первого объекта.
11. Вывод содержимого массива второго объекта.
12. Вывод суммы элементов массива второго объекта.
13. Второму объекту присвоить первый объект.
14. С использованием первого указателя вызов метода 1.
15. Вывод содержимого массива второго объекта.
16. Вывод суммы элементов массива второго объекта.
17. Удалит первый объект.
18. Удалить второй объект.

Добавить в этот алгоритм пункты, которые обеспечат корректное завершение работы программы.

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
Constructor set
Copy constructor
20  5  4  2
31
100  5  8  2
```

```
115
100  5  8  2
115
Destructor
Destructor
```

2 МЕТОД РЕШЕНИЯ

Класс cl1:

- свойства/поля:
  - поле для объявления указателя на динамический массив:
    - наименование — array;
    - тип — int*;
    - модификатор доступа — private;
  - поле для хранения размера массива array (скрытого поля класса cl1)):
    - наименование — arraysize;
    - тип — int;
    - модификатор доступа — private;
- функционал:
  - метод getter — возвращение значения указателя на массив из закрытой области;
  - метод setter — присваивание значения указателя на массиву из закрытой области.

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 метода getter класса cl1

Функционал: возвращение значения указателя на массив из закрытой области.

Параметры: нет.

Возвращаемое значение: int*, указатель на массив.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода getter класса cl1

№	Предикат	Действия	№ перехода
1		Возврат значения поля array текущего объекта	∅

3.2 Алгоритм метода setter класса cl1

Функционал: присваивание значения указателя на массиву из закрытой области.

Параметры: нет.

Возвращаемое значение: int* array, указатель на новое значение для поля array текущего объекта.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода setter класса cl1

№	Предикат	Действия	№ перехода
1		Присваивание полю array текущего объекта значение параметра array	∅

3.3 Алгоритм функции main

Функционал: главная функция программы.

Параметры: нет.

Возвращаемое значение: целое, индикация корректности работы программы.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленной переменной arraysize	2
2		Ввод значения переменной arraysize с клавиатуры	3
3	Значение переменной arraysize больше 2 и чётное	Вывод: <<значение arraysize>>	4
	Значение переменной arraysize не больше 2 и не чётное	Вывод: <<значение arraysize>>"?"	∅
4		Создание указателя на объект Object1 класса cl1	5
5		Присваивание объекту Object1 возвращаемого значения функции function с аргументом arraysize	6
6		Вызов метода SummBy1() объекта Object1	7
7		Создание указателя на объект Object2 класса cl1 и выделение под него памяти с помощью оператора new	8
8		Вызов метода MultiplyBy1() объекта Object2	9
9		Вызов метода OutputArray() объекта Object2	10
10		Вывод: <<результат вызова метода SummAll()	11

№	Предикат	Действия	№ перехода
		объекта Object2>>	
11		Вызов метода OutputArray() объекта Object2	12
12		Вывод: <<результат вызова метода SummAll() объекта Object2>>	13
13		Инициализация указателя ptrarr возвращаемым значением метода getter() объекта Object2	14
14		Присваивание объекту Object2 значение объекта Object1	15
15		Вызов метода MultiplyBy1() объекта Object1	16
16		Вызов функции setter() объекта Object2 с передачей ей значения указателя ptrarr	17
17		Вызов метода OutputArray() объекта Object2	18
18		Вывод: <<результат вызова метода SummAll() объекта Object2>>	19
19		Удаление объекта Object1 посредством оператора delete	20
20		Удаление объекта Object2 посредством оператора delete	∅

3.4 Алгоритм функции function

Функционал: Создание локального объета с использованием параметризированного конструктора.

Параметры: arraysize, параметр для задавания размера массива создаваемого объекта.

Возвращаемое значение: cl1*, указатель на созданный объект класса cl1.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции function

№	Предикат	Действия	№ перехода
1		Создание указателя на объект Object3 класса cl1 и выделение под него памяти с помощью оператора new	2
2		Вызов метода CreateArray() объекта Object3	3
3		Вызов метода fillarray() объекта Object3	4
4		Вызов метода MultiplyBy1() объекта Object3	5
5		Возврат указателя на объект Object3	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-2.

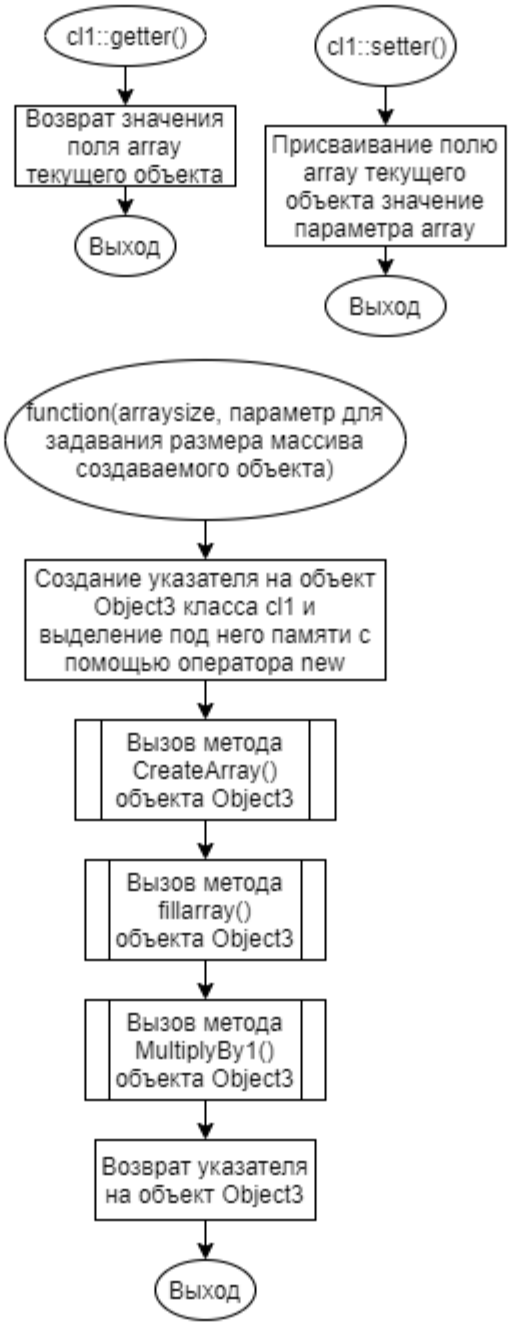

Рисунок 1 – Блок-схема алгоритма

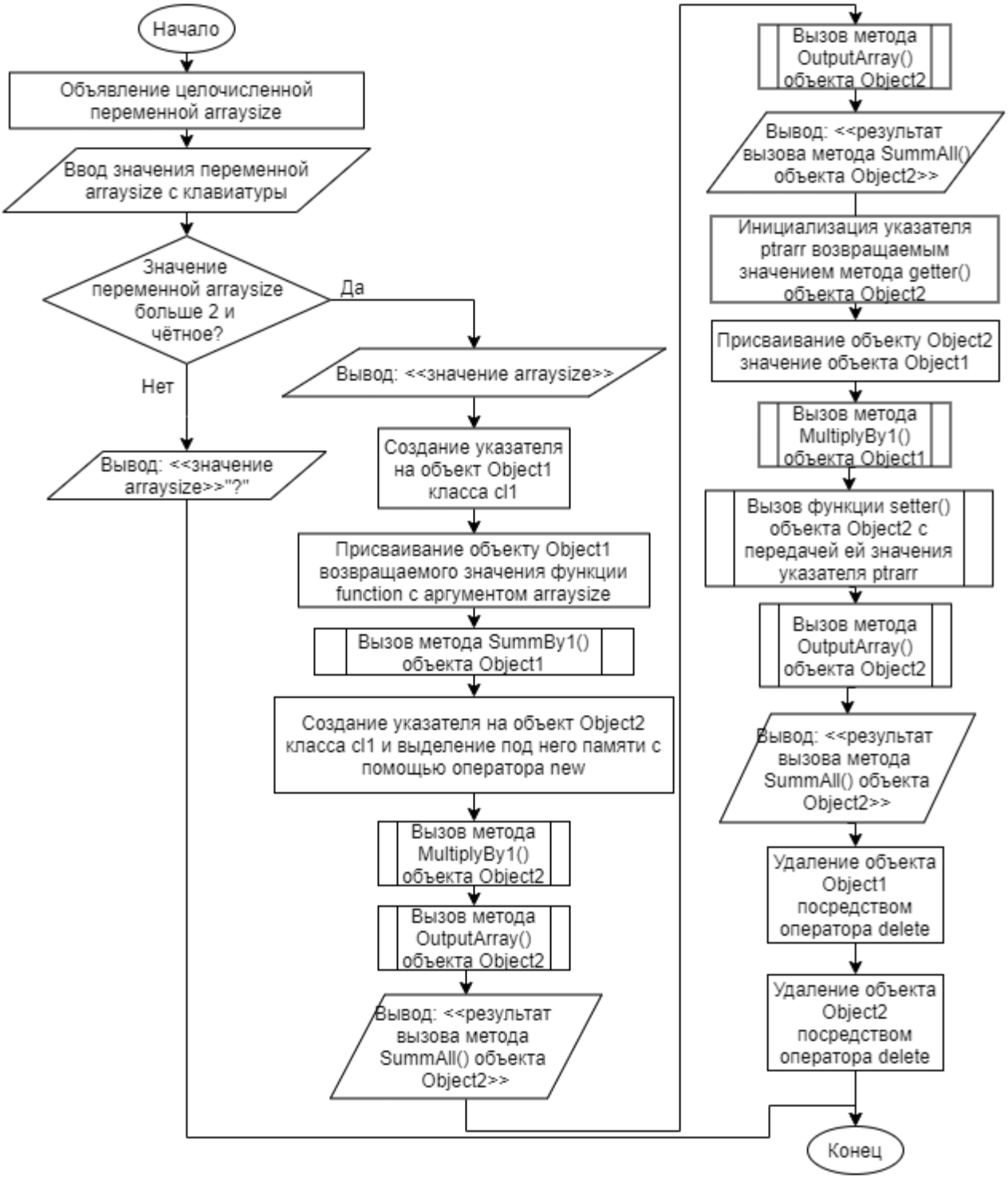

Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл cl1.cpp

Листинг 1 – cl1.cpp

```
#include <iostream>
#include "cl1.h"
using namespace std;
cl1::cl1()
{
   cout << "Default constructor" << endl;
}
cl1::cl1(int sizeofarray)
{
   cout << "Constructor set";
   this -> sizeofarray = sizeofarray;
}
cl1::cl1(const cl1 &Object1)
{
   cout << "\nCopy constructor" << endl;
   sizeofarray = Object1.sizeofarray;
   array = new int[sizeofarray];
   for (int i = 0; i < sizeofarray; i++)
   {
      array[i] = Object1.array[i];
   }
}
cl1::~cl1()
{
   cout << "\nDestructor";
}
void cl1::fillarray()
{
   for (int i = 0; i < sizeofarray; i++)
   {
      cin >> array[i];
   }
}
int cl1::SummAll()
{
   int summ = 0;
   for (int i = 0; i < sizeofarray; i++)
   {
      summ += array[i];
```

```
   }
   return summ;
}
void cl1::SummBy1()
{
   for (int i = 0; i < sizeofarray; i += 2)
   {
      array[i] += array[i+1];
   }
}
void cl1::MultiplyBy1()
{
   for (int i = 0; i < sizeofarray; i+= 2)
   {
      array[i] *= array[i+1];
   }
}
void cl1::CreateArray()
{
   array = new int[sizeofarray];
}
void cl1::OutputArray()
{
   cout << array[0];
   for (int i = 1; i < sizeofarray; i++)
   {
      cout << "   " << array[i];
   }
   cout << endl;
}
int* cl1::getter()
{
   return array;
}
void cl1::setter(int* array)
{
   this -> array = array;
}
```

5.2 Файл cl1.h

Листинг 2 – cl1.h

```
#ifndef __CL1__H
#define __CL1__H
class cl1
{
   int* array;
   int sizeofarray;
   public:
```

```
        cl1();
        cl1(int sizeofarray);
        cl1(const cl1 &Object1);
        ~cl1();
        void fillarray();
        int SummAll();
        void SummBy1();
        void MultiplyBy1();
        void CreateArray();
        void OutputArray();
        int* getter();
        void setter(int*);
    };
    #endif
```

5.3 Файл main.cpp

Листинг 3 – main.cpp

```
    #include <iostream>
    #include "cl1.h"
    using namespace std;

    cl1* function(int arraysize)
    {
       cl1* Object3 = new cl1(arraysize);
       Object3 -> CreateArray();
       Object3 -> fillarray();
       Object3 -> MultiplyBy1();
       return Object3;
    }


    int main()
    {
       int arraysize;
       cin >> arraysize;
       if(arraysize > 2 && arraysize%2 == 0)
       {
          cout << arraysize << endl;
          cl1* Object1;
          Object1 = function(arraysize);
          Object1 -> SummBy1();
          cl1* Object2 = new cl1(*Object1);
          Object2 -> MultiplyBy1();
          Object1 -> OutputArray();
          cout << Object1 -> SummAll() << endl;
          Object2 -> OutputArray();
          cout << Object2 -> SummAll() << endl;
          int* ptrarr = Object2 -> getter();
```

```
            *Object2 = *Object1;
            Object1 -> SummBy1();
            Object2 -> setter(ptrarr);
            Object2 -> OutputArray();
            cout << Object2 -> SummAll();
            delete Object1;
            delete Object2;
         }
         else
         {
            cout << arraysize << "?";
         }
         return(0);
      }
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 5.

Таблица 5 – Результат тестирования программы

Входные данные	**Ожидаемые выходные данные**	**Фактические выходные данные**
4 1 2 3 4	4 Constructor set Copy constructor 4 2 16 4 26 8 2 64 4 78 8 2 64 4 78 Destructor Destructor	4 Constructor set Copy constructor 4 2 16 4 26 8 2 64 4 78 8 2 64 4 78 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. 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).