

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	6
1.2 Описание выходных данных.....	7
2 МЕТОД РЕШЕНИЯ.....	9
3 ОПИСАНИЕ АЛГОРИТМОВ.....	10
3.1 Алгоритм функции main.....	10
3.2 Алгоритм функции Func.....	10
3.3 Алгоритм метода In класса Class.....	11
3.4 Алгоритм метода Out класса Class.....	11
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	13
5 КОД ПРОГРАММЫ.....	16
5.1 Файл Class.cpp.....	16
5.2 Файл Class.h.....	17
5.3 Файл main.cpp.....	18
6 ТЕСТИРОВАНИЕ.....	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	20

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;
- Параметризованный 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:

1. Создание локального объекта с использованием параметризованного конструктора.
2. Возврат созданного локального объекта.

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. Создание первого объекта.
5. Присвоение первому объекту результата работы функции func с аргументом, содержащим значение размерности массива.
6. Для первого объекта вызов метода создания массива.
7. Для первого объекта вызов метода ввода данных массива.
8. Для первого объекта вызов метода 2.
9. Инициализация второго объекта первым объектом.
10. Вызов метода 1 для второго объекта.
11. Вывод содержимого массива первого объекта.
12. Вывод суммы элементов массива первого объекта.
13. Вывод содержимого массива второго объекта.
14. 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

Параметризованный 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:

- объект `a` класса `Class` предназначен для первый объект;
- объект `b` класса `Class` предназначен для второй объект;
- объект `a_local` класса `Class` предназначен для локальный объект.

Класс `Class`:

- функционал:
 - метод `In` — выделение памяти для целочисленного массива длиной `n`;
 - метод `Out` — вывод элементов массива на экран.

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: нет.

Возвращаемое значение: целочисленное - индикатор корректности выполнения алгоритма.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление объекта a класса Class	2
2		присваивание a = Func с параметром n1	3
3		вызов метода In для объекта a с параметром n1	4
4		вызов метода Input для объекта a	5
5		вызов метода Method2 для объекта a	6
6		инициализация объекта b класса Class = a	7
7		вызов метода Method1 для объекта b	8
8		вызов метода Out для объекта a	9
9		вывод с новой строки метода Sum для объекта a	10
10		вызов метода Out для объекта b	11
11		вывод с новой строки метода Sum для объекта b	Ø

3.2 Алгоритм функции Func

Функционал: Создание локального объекта.

Параметры: int n1 - длина массива mas.

Возвращаемое значение: Class - возвращает объект класса Class.

Алгоритм функции представлен в таблице 2.

Таблица 2 – Алгоритм функции Func

№	Предикат	Действия	№ перехода
1		объявление объекта a_local класса Class с параметром n1	2
2		возвращение a_local	Ø

3.3 Алгоритм метода In класса Class

Функционал: выделение памяти для целочисленного массива длиной n.

Параметры: int n1 - длина массива.

Возвращаемое значение: void - не возвращает значений.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода In класса Class

№	Предикат	Действия	№ перехода
1		присвоение n = n1	2
2		выделение памяти для mas = new int[n]	Ø

3.4 Алгоритм метода Out класса Class

Функционал: вывод элементов массива на экран.

Параметры: нет.

Возвращаемое значение: void - не возвращает значений.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода Out класса Class

№	Предикат	Действия	№ перехода
1		инициализация переменной счетчика $i=0$	2
2	$i < n$	вывод с новой строки mas[i] с разделителем в 3 пробела; инкремент i	2
			3
3		вывод переноса строки	\emptyset

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

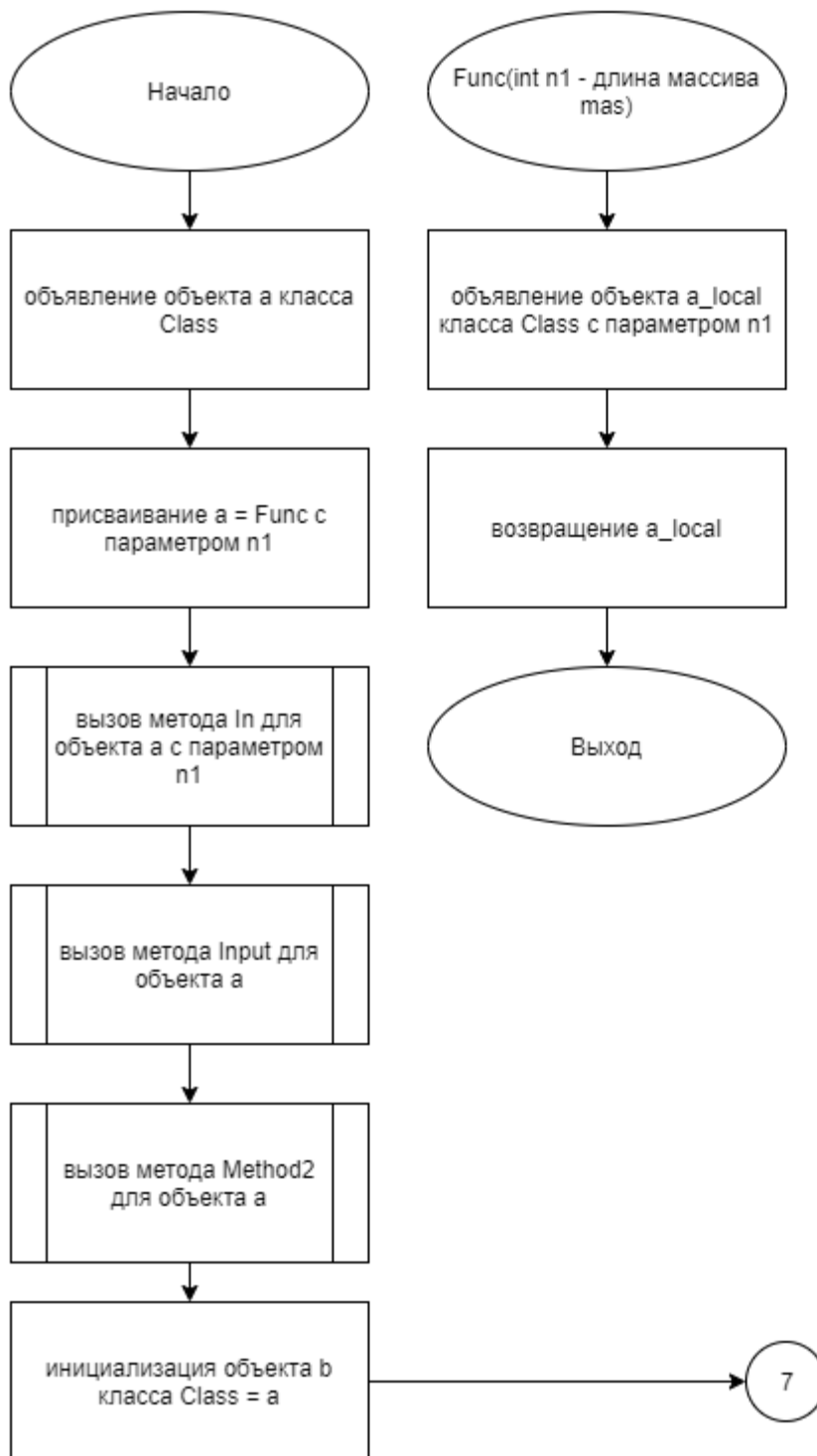


Рисунок 1 – Блок-схема алгоритма

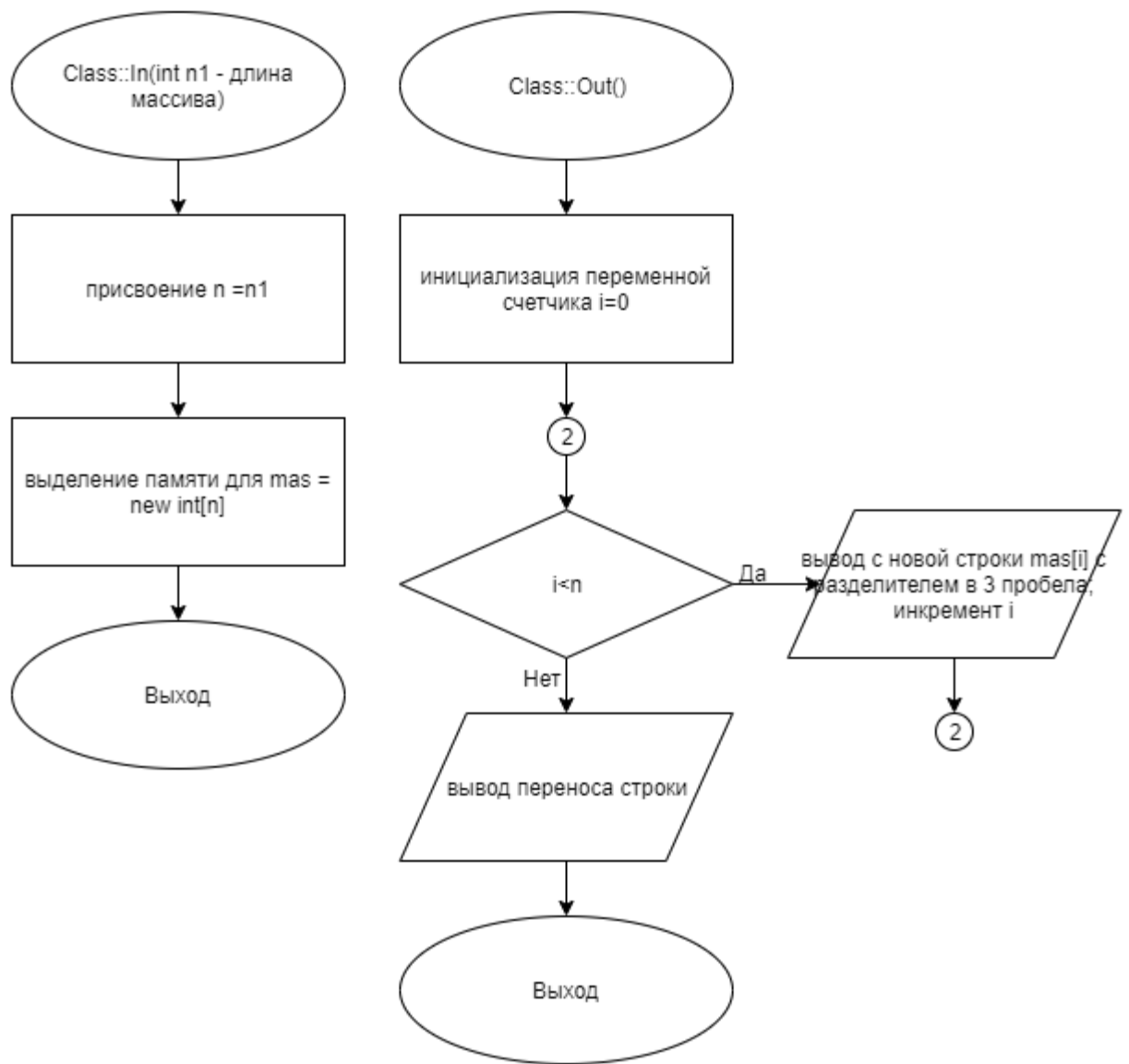


Рисунок 2 – Блок-схема алгоритма



Рисунок 3 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

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 n1){
    cout<<"Constructor set"<<endl;
}

Class::Class(const Class & a){
    cout<<"Copy constructor"<<endl;
    n=a.n;
    mas=new int [n];
    for (int i =0;i<n;i++){
        mas[i]=a.mas[i];
    }
}

void Class::In(int n1){
    n=n1;
    mas=new int[n];
}

void Class::Input(){
    for (int i=0;i<n;i++){
        cin>>mas[i];
    }
}

int Class::Method1(){
    int s=0;
    for (int i=0;i<n;i+=2){
        mas[i]=mas[i]+mas[i+1];
        s+=mas[i]+mas[i+1];
    }
    return s;
}

int Class::Method2(){
    int s=0;
    for (int i=0;i<n;i+=2){
        mas[i]=mas[i]*mas[i+1];
    }
}
```

```

        s+=mas[i]+mas[i+1];
    }
    return s;
}
int Class::Sum(){
    int sum=0;
    for (int i=0;i<n;i++){
        sum+=mas[i];
    }
    return sum;
}
void Class::Out(){
    for (int i=0; i< n; i++){
        cout<< mas[i];
        if (i!=n-1){
            cout<<" ";
        }
    }
    cout<<endl;
}
Class::~~Class(){
    cout<<"Destructor"<<endl;
}

```

5.2 Файл Class.h

Листинг 2 – Class.h

```

#ifndef __CLASS__H
#define __CLASS__H
class Class{
    private:
        int* mas;
        int n;
    public:
        Class();
        Class(int n1);
        Class(const Class & a);
        void In(int n1);
        void Input();
        int Method1();
        int Method2();
        int Sum();
        void Out();
        ~Class();

};

#endif

```

5.3 Файл main.cpp

Листинг 3 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include "Class.h"
#include <iostream>
using namespace std;
Class Func(int n1){
    Class a_local(n1);
    return a_local;
}
int main()
{
    int n1;
    cin>>n1;
    if (n1<=2 || n1%2!=0){
        cout<<n1<<"?";
        return(0);
    }
    cout<<n1<<endl;
    Class a;
    a=Func(n1);
    a.In(n1);
    a.Input();
    a.Method2();
    Class b = a;
    b.Method1();
    a.Out();
    cout<<a.Sum()<<endl;
    b.Out();
    cout<<b.Sum()<<endl;
    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 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-avvora.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-avvora.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).