

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	7
1.2 Описание выходных данных.....	8
2 МЕТОД РЕШЕНИЯ.....	10
3 ОПИСАНИЕ АЛГОРИТМОВ.....	11
3.1 Алгоритм метода GetArr класса MyClass.....	11
3.2 Алгоритм метода SetArr класса MyClass.....	11
3.3 Алгоритм функции function.....	12
3.4 Алгоритм функции main.....	12
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	14
5 КОД ПРОГРАММЫ.....	16
5.1 Файл main.cpp.....	16
5.2 Файл MyClass.cpp.....	17
5.3 Файл MyClass.h.....	18
6 ТЕСТИРОВАНИЕ.....	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	21

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;
- параметризованный конструктор, передается целочисленный параметр. Параметр должен иметь значение больше 2 и быть четным. В начале работы выдает сообщение;
- конструктор копии, обеспечивает создание копии объекта в новой области памяти. В начале 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 параметризованного 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

Для решения задачи используется:

- для решения задачи используется то же , что и в предыдущей задаче 3_1_4, однако класс MyClass , функции main и function переработаны под условия задачи.

Класс MyClass:

- свойства/поля:
 - поле arr и ArrSize класса MyClass остались неизменными:
 - наименование — ;
 - тип — ;
 - модификатор доступа — private;
- функционал:
 - метод те же методы , что и в предыдущей задаче 3_1_4 — ;
 - метод GetArr — используется для возвращения значения указателя на массив из закрытой области;
 - метод SetArr — используется для присваивания значения указателя 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 GetArr класса MyClass

Функционал: используется для возвращения значения указателя на массив из закрытой области.

Параметры: нет.

Возвращаемое значение: `int*` - значение указателя на массив.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода GetArr класса MyClass

№	Предикат	Действия	№ перехода
1		возврат значения поля <code>arr</code> текущего объекта	Ø

3.2 Алгоритм метода SetArr класса MyClass

Функционал: используется для присваивания значения указателя массиву из закрытой области.

Параметры: `int*` `arr` - указатель на новое значение для поля `arr` текущего объекта.

Возвращаемое значение: `void`.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода SetArr класса MyClass

№	Предикат	Действия	№ перехода
1		Присваивание полю arr текущего объекта значение параметра arr	Ø

3.3 Алгоритм функции function

Функционал: Создание локального объекта с использованием параметризованного конструктора.

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

Возвращаемое значение: MyClass* - возврат указателя на созданный объект класса MyClass.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции function

№	Предикат	Действия	№ перехода
1		создание указателя на объект obj класса MyClass и выделение под него памяти с помощью оператора new	2
2		вызов метода GreateArray объекта obj	3
3		вызов метода FillArray объекта obj	4
4		вызов метода Method2 объекта obj	5
5		возврат указателя на объект obj	Ø

3.4 Алгоритм функции main

Функционал: Основной алгоритм программы.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 4.

Таблица 4 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		начало работы алгоритма из предыдущей задачи 3_1_4	2
2		создание указателя на объект obj1 класса MyClass	3
3		присваивание объекту obj1 результата функции function с аргументом size	4
4		вызов метода Method1 объекта obj1	5
5		Создание указателя на объект obj2 класса MyClass и выделение под него памяти с помощью оператора new	6
6		вызов метода Method2 объекта obj2	7
7		вызов метода OutputArray объекта obj1	8
8		вывод результата вызова метода Sum объекта obj1 на экран	9
9		вызов метода OutputArray объекта obj2	10
10		вывод результат вызова метода Sum объекта obj2 на экран	11
11		присваивание объекта obj2 значение объекта obj1	12
12		вызов метода Method2 объекта obj1	13
13		вызов метода OutputArray объекта obj2	14
14		вывод результата вызова метода Sum объекта obj2 на экран	15
15		Удаление объекта obj1 с помощью оператора delete	16
16		удаление объекта obj1 с помощью оператора delete	17
17		Окончание работы алгоритма ищ предыдущей задачи 3_1_4	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

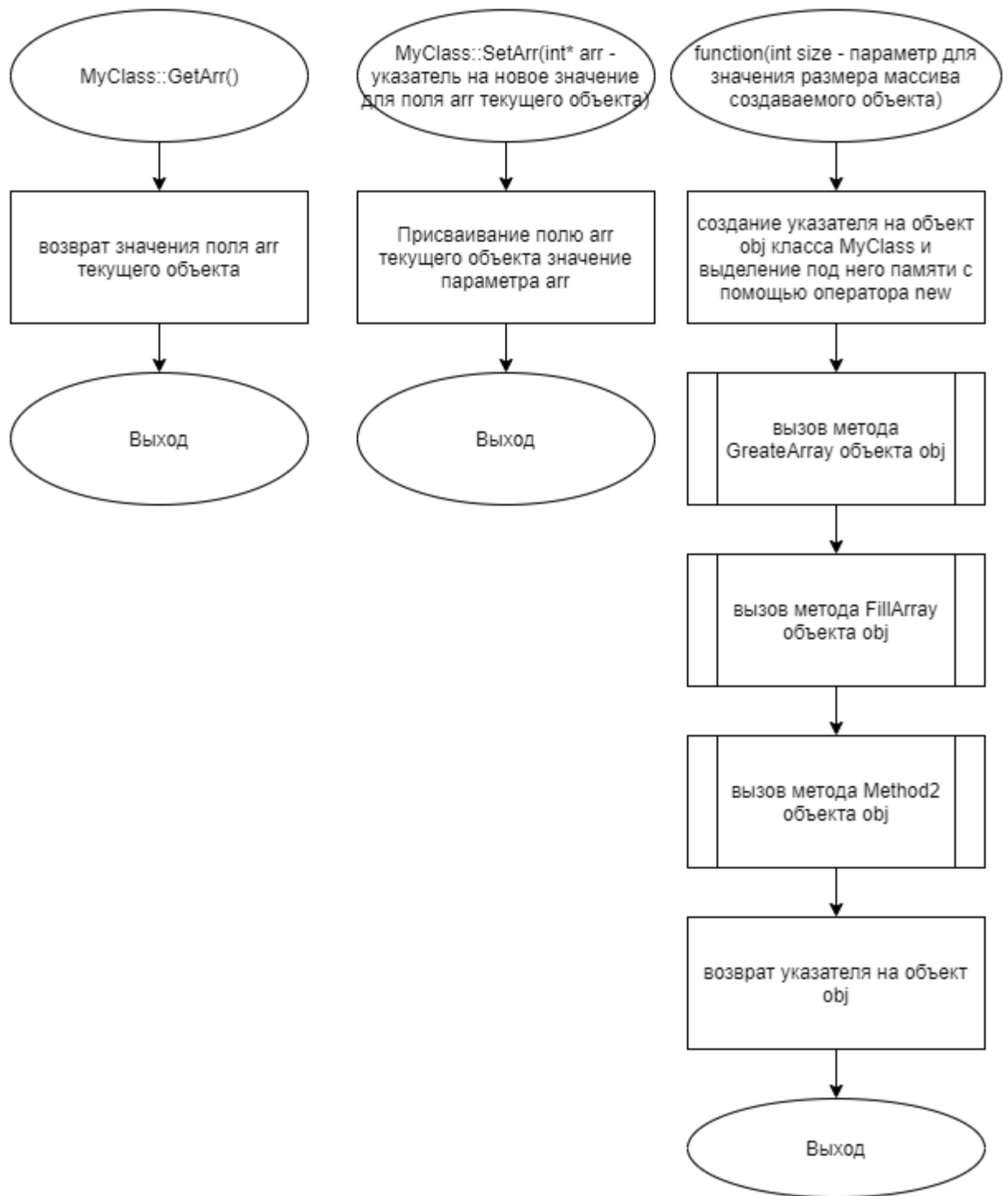


Рисунок 1 – Блок-схема алгоритма

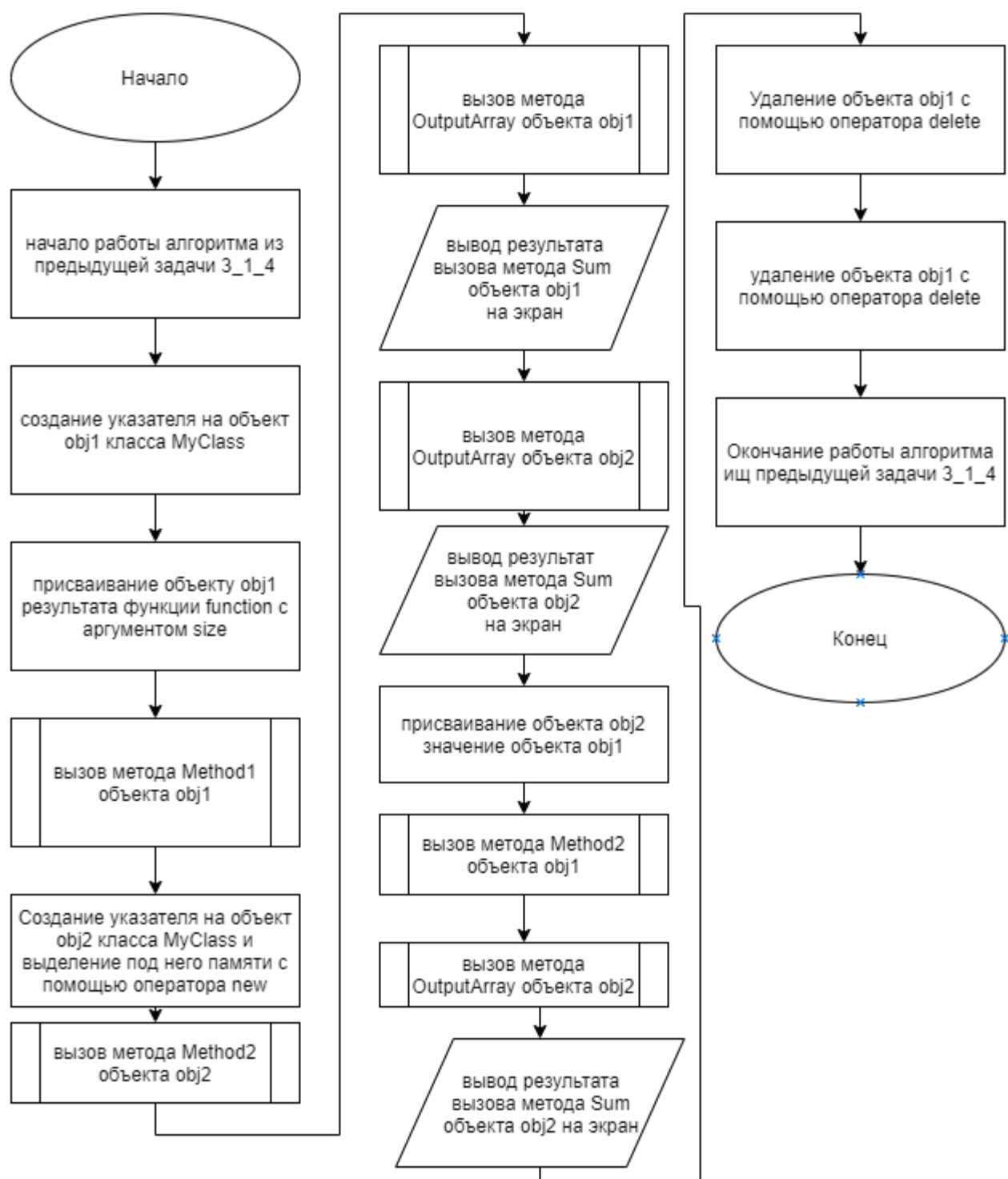


Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include <iostream>
#include "MyClass.h"
using namespace std;
MyClass* function(int size)
{
    MyClass* obj = new MyClass(size);
    obj->CreateArray();
    obj->FillArray();
    obj->Method2();
    return obj;
}

int main(){
    int size;
    cin >>size;
    if (size >2 && size%2 == 0)
    {
        cout << size <<endl;
        MyClass* obj1;
        obj1 = function(size);
        obj1->Method1();
        MyClass* obj2 = new MyClass(*obj1);
        obj2->Method2();
        obj1->OutputArray();
        cout << obj1->Sum() << endl;
        obj2->OutputArray();
        cout << obj2->Sum() << endl;
        *obj2 = *obj1;
        obj1->Method2();
        obj2->OutputArray();
        cout << obj2->Sum();
        delete obj1;
        delete obj2;
    }
    else
    {
        cout << size << "?";
    }
    return (0);
}
```

```
}
```

5.2 Файл MyClass.cpp

Листинг 2 – MyClass.cpp

```
#include "MyClass.h"
MyClass::MyClass()
{
    cout << "Default constructor" << endl;
}
MyClass::MyClass(int ArrSize)
{
    cout << "Constructor set";
    this->ArrSize = ArrSize;
}
MyClass::MyClass(const MyClass &obj)
{
    cout << "\nCopy constructor" << endl;
    ArrSize = obj.ArrSize;
    arr = new int [ArrSize];
    for (int i = 0; i < ArrSize; i++)
    {
        arr[i] = obj.arr[i];
    }
}

MyClass::~MyClass()
{
    cout << "\nDestructor";
}
void MyClass::FillArray()
{
    for (int i=0 ; i < ArrSize; i++)
    {
        cin >> arr[i];
    }
}
void MyClass::CreateArray()
{
    arr = new int [ArrSize];
}
void MyClass::OutputArray()
{
    cout << arr[0];
    for (int i = 1; i < ArrSize; i++)
    {
        cout << " " << arr[i];
    }
    cout << endl;
}
```

```

}
int* MyClass::GetArr()
{
    return arr;
}
void MyClass::SetArr(int* arr)
{
    this->arr=arr;
}
int MyClass::Sum()
{
    int sum = 0;
    for (int i=0; i < ArrSize; i++)
    {
        sum+=arr[i];
    }
    return sum;
}

void MyClass::Method1()
{
    for (int i =0; i< ArrSize; i+=2)
    {
        arr[i] +=arr[i+1];
    }
}
void MyClass::Method2()
{
    for (int i =0; i < ArrSize; i+=2)
    {
        arr[i] *= arr[i+1];
    }
}

```

5.3 Файл MyClass.h

Листинг 3 – MyClass.h

```

#ifndef __MYCLASS__H
#define __MYCLASS__H
#include <iostream>
using namespace std;
class MyClass
{
    int* arr;
    int ArrSize;

public:
    MyClass();
    MyClass(int ArrSize);

```



```
    MyClass(const MyClass &obj);  
    ~MyClass();  
    void FillArray();  
    void CreateArray();  
    void OutputArray();  
    int* GetArr();  
    void SetArr(int*);  
    int Sum();  
    void Method1();  
    void Method2();  
};  
  
#endif
```

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 Constructor set Copy constructor 20 5 4 2 31 100 5 8 2 115 100 5 8 2 115 Destructor Destructor	4 Constructor set Copy constructor 20 5 4 2 31 100 5 8 2 115 100 5 8 2 115 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).