



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

« МИРЭА Российский технологический университет »

РТУ МИРЭА

Институт Информационных технологий

Кафедра Вычислительной техники

УЧЕБНОЕ ЗАДАНИЕ

по дисциплине

« Объектно-ориентированное программирование »

Наименование задачи:

« Задача 4_3_1 »

С тудент группы

ИКБО-13-21

Черномуров С.А.

Руководитель практики

Ассистент

Асадова Ю.С.

Работа представлена

«__»_____ 2022 г.

(подпись студента)

Оценка

(подпись руководителя)

Москва 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
Постановка задачи.....	5
Метод решения.....	7
Описание алгоритма.....	10
Блок-схема алгоритма.....	15
Код программы.....	19
Тестирование.....	22
ЗАКЛЮЧЕНИЕ.....	23
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ).....	24

ВВЕДЕНИЕ

Постановка задачи

Создать базовый класс, содержащий только свойства. На его базе создать производный класс 1 и производный класс 2, содержащие только методы. В первом производном классе реализовать метод ввода размерности целочисленного массива и значений элементов массива. Во втором классе реализовать метод упорядочения целочисленного массива по не убыванию и метод вывода массива.

Посредством множественного наследования классов 1 и 2 создать класс 3. В классе 3 реализовать метод с алгоритмом:

1. Вызов метода ввода целочисленного массива.
2. Вызов метода вывода массива.
3. Вызов метода упорядочения целочисленного массива.
4. Вызов метода вывода массива.

Описание входных данных

Первая строка:

«размерность целочисленного массива»

Размерность массива натуральное число больше или равно 1 и меньше или равно 100.

Вторая строка:

«последовательность целых чисел»

Количество целых чисел равно размерности массива и разделены пробелами. Значение каждого числа принадлежит интервалу $[-100, 100]$.

Описание выходных данных

Первая строка:

Array dimension: «размерность целочисленного массива»

Вторая

строка:

The original array: «последовательность целых чисел»

Для вывода каждого числа выделяется поле из 5 позиции.

Третья

строка:

An ordered array: «последовательность целых чисел»

Для вывода каждого числа выделяется поле из 5 позиции.

Метод решения

Для решения задачи используются:

- Объекты стандартных потоков ввода и вывода `cin` и `cout` соответственно. Используются для ввода с клавиатуры и вывода на экран.
- Функция `swap`, принадлежит библиотеке `algorithm`. Используется для сортировки массива (обмена значений элементов массива).
- Манипулятор потока ввода/вывода `setw`, принадлежит библиотеке `iomanip`. Используется для форматирования вывода.
- Объект `obj` класса `Cl3`. Используется для создания объекта.
- Объекты классов `Base`, `Cl1`, `Cl2`.
- **Класс `Base`:**
 - Свойства/поля:
 - Свойство:
 - Наименование - `arr`;
 - Тип - указатель на целочисленный массив;
 - Модификатор доступа - защищенный.
 - Свойство:
 - Наименование - `len`;
 - Тип - целочисленный;
 - Модификатор доступа - защищенный.
- **Класс `Cl1`:**
 - Методы:
 - Метод `Fill`:
 - Функционал - метод ввода размерности целочисленного массива и значений элементов массива.

- **Класс C12:**
 - Методы:
 - Метод Sort:
 - Функционал - метод упорядочения целочисленного массива по не убыванию.
 - Метод Print:
 - Функционал - метод вывода массива.
- **Класс C13:**
 - Методы:
 - Метод Method:
 - Функционал - метод, вызывающий методы инициализации массива, его сортировки и его вывода.

Иерархия наследования:

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер	Комментарий
1	Base			Базовый класс, содержащий только свойства		
		C11	virtual public		2	
		C12	virtual public		3	
2	C11			Производный класс класса Base, является головным для класса C13		
		C13	public		4	

3	Cl2			Производный класс класса Base, является головным для класса Cl3		
		Cl3	public		4	
4	Cl3			Производный класс классов Cl1 и Cl2		

Описание алгоритма

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

Функция: main

Функционал: Основной алгоритм программы

Параметры: Отсутствуют

Возвращаемое значение: Целочисленный тип данных - код возврата

Алгоритм функции представлен в таблице 2.

Таблица 2. Алгоритм функции main

№	Предикат	Действия	№ перехода	Комментарий
1		Создание объекта obj класса Cl3	2	
2		Вызова метода Method объекта obj	Ø	

Класс объекта: Cl1

Модификатор доступа: public

Метод: Fill

Функционал: Метод ввода размерности целочисленного массива и значений элементов массива

Параметры: Отсутствуют

Возвращаемое значение: Отсутствует

Алгоритм метода представлен в таблице 3.

Таблица 3. Алгоритм метода Fill класса Cl1

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление целочисленной переменной len	2	
2		Считывание с клавиатуры значения переменной len	3	
3		Присвоение значения len свойству len объекта класса Base	4	
4		Динамическое выделение памяти под целочисленный массив arr объекта класса Base длиной len	5	
5		Объявление целочисленной переменной с инициализацией i=0	6	Использование i в качестве счетчика для цикла
6	Значение i меньше значения свойства len объекта класса Base	Считывание с клавиатуры значения arr[i] объекта класса Base	7	
			8	Выход из цикла
7		Инкрементирование i	6	
8		Вывод на экран "Array dimension: ", Значение свойства len объекта класса Base с последующим переносом на новую строку	9	
9		Вывод на экран "The original array:"	Ø	

Класс объекта: Cl2

Модификатор доступа: public

Метод: Sort

Функционал: Метод упорядочения целочисленного массива по не убыванию

Параметры: Отсутствуют

Возвращаемое значение: Отсутствует

Алгоритм метода представлен в таблице 4.

Таблица 4. Алгоритм метода Sort класса Cl2

№	Предикат	Действия	№ перехода	Комментарий
1		Вывод на экран переноса на новую строку	2	
2		Объявление целочисленной переменной с инициализацией $i=0$	3	Использование i в качестве счетчика для цикла
3	Значение i меньше значения декрементированного свойства len объекта класса Base	Объявление целочисленной переменной с инициализацией $j=i+1$	5	Использование j в качестве счетчика для вложенного цикла
			8	Выход из цикла
4	Значение j меньше значения свойства len объекта класса Base		5	
			7	Выход из вложенного цикла
5	Значение arr[i] объекта класса Base	Вызов функции swap с	6	

	больше значения arr[j] объекта класса Base	целочисленными параметрами: элемент массива arr[i] объекта класса Base, элемент массива arr[j] объекта класса Base		
			6	
6		Инкрементирование j	4	
7		Инкрементирование i	3	
8		Вывод на экран "An ordered array:"	Ø	

Класс объекта: Cl2

Модификатор доступа: public

Метод: Print

Функционал: Метод вывода массива

Параметры: Отсутствуют

Возвращаемое значение: Отсутствует

Алгоритм метода представлен в таблице 5.

Таблица 5. Алгоритм метода Print класса Cl2

№	Предикат	Действия	№ перехода	Комментарий
1		Объявление целочисленной переменной с инициализацией i=0	2	Использование i в качестве счетчика для цикла
2	Значение i меньше значения	Вывод на экран значения arr[i] объекта класса Base с длиной поля вывода 5	3	

	свойства len объекта класса Base	позиций		
			∅	Выход из цикла
3		Инкрементирование i	2	

Класс объекта: C13

Модификатор доступа: public

Метод: Method

Функционал: Метод, вызывающий методы инициализации массива, его сортировки и его вывода

Параметры: Отсутствуют

Возвращаемое значение: Отсутствует

Алгоритм метода представлен в таблице 6.

Таблица 6. Алгоритм метода Method класса C13

№	Предикат	Действия	№ перехода	Комментарий
1		Вызов метода Fill объекта класса C11	2	
2		Вызов метода Print объекта класса C12	3	
3		Вызов метода Sort объекта класса C12	4	
4		Вызов метода Print объекта класса C12	∅	

Блок-схема алгоритма

Представим описание алгоритмов в графическом виде на рисунках ниже.

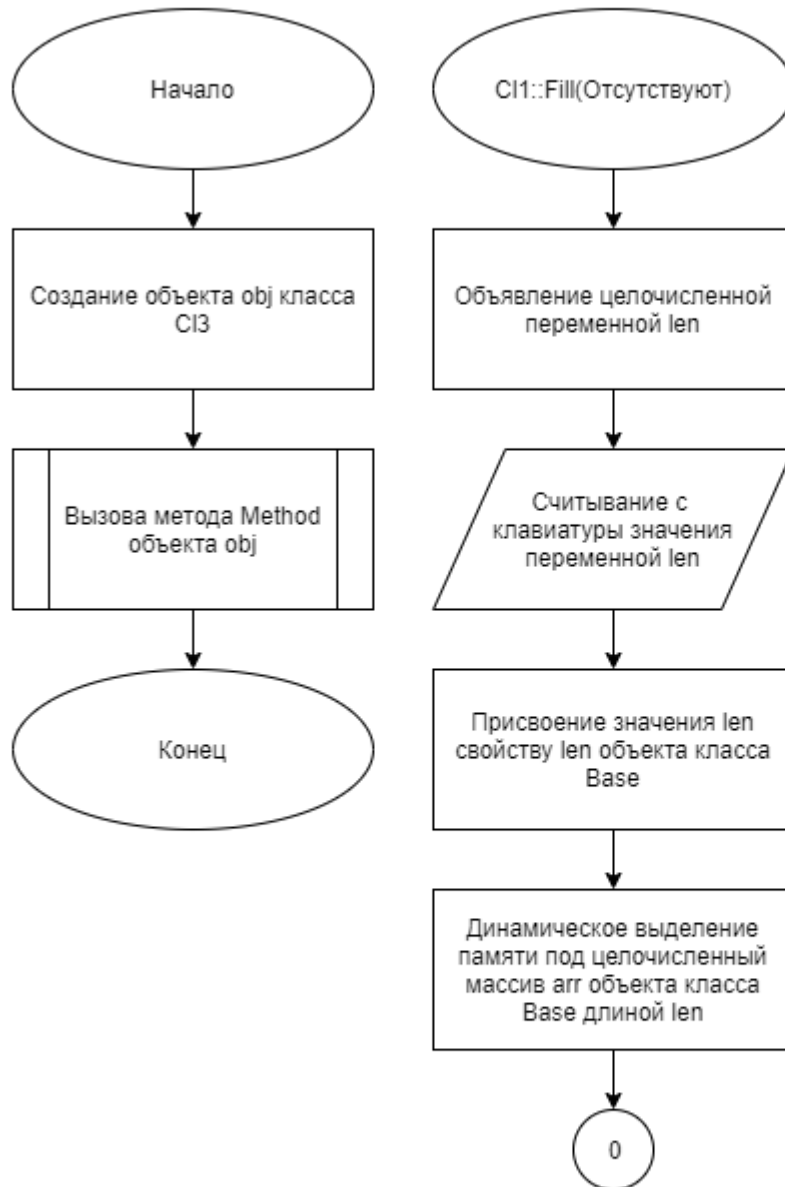


Рис. 1. Блок-схема алгоритма.

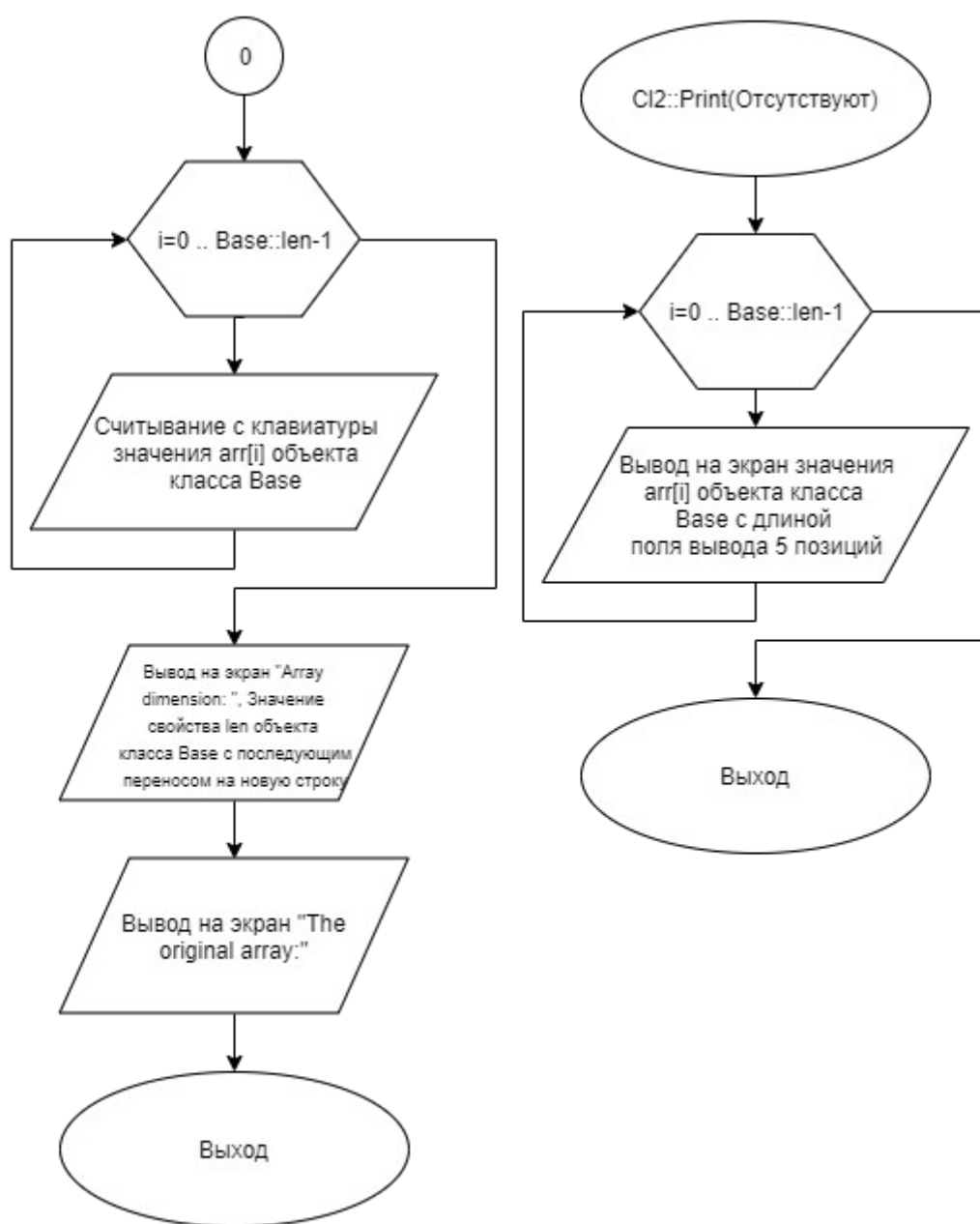


Рис. 2. Блок-схема алгоритма.

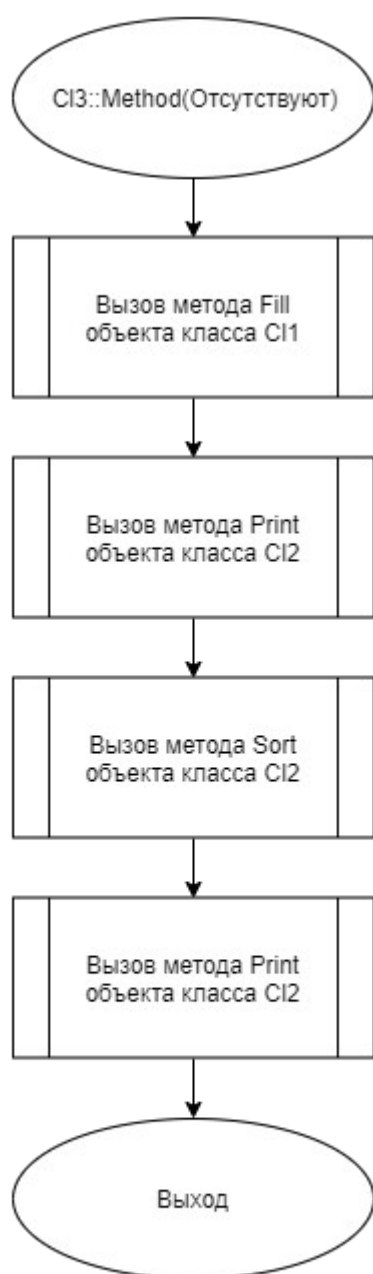


Рис. 3. Блок-схема алгоритма.

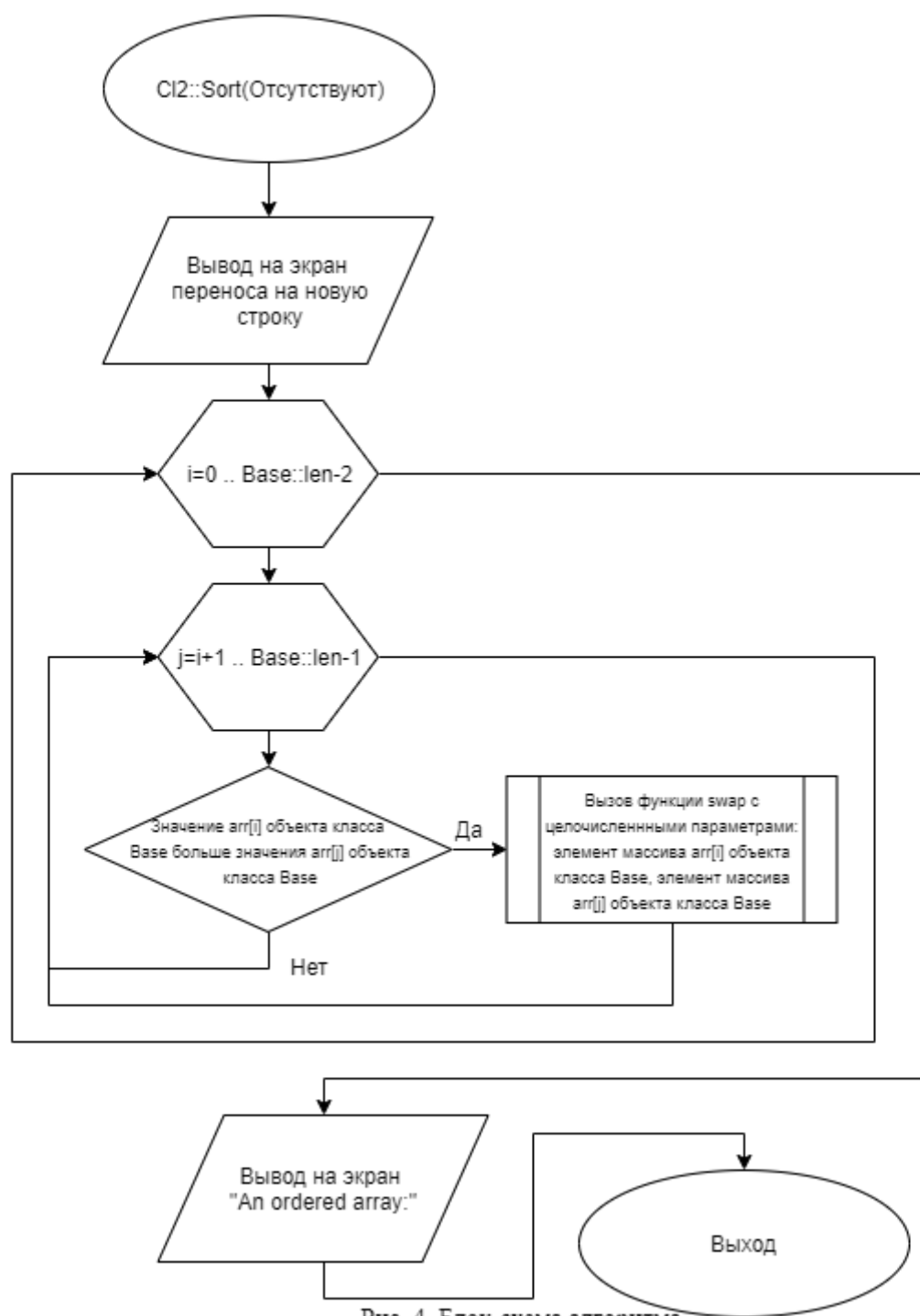


Рис. 4. Блок-схема алгоритма.

Код программы

Программная реализация алгоритмов для решения задачи представлена ниже.

Файл Base.h

```
#ifndef BASE_H
#define BASE_H

class Base{
protected:
    int* arr;
    int len;
};

#endif
```

Файл Cl1.cpp

```
#include "Cl1.h"
#include <iostream>
using namespace std;

void Cl1 :: Fill(){
    int len;
    cin >> len;
    Base :: len = len;
    Base :: arr = new int[Base :: len];
    for (int i=0; i<Base :: len; i++){
        cin >> Base :: arr[i];
    }
    cout << "Array dimension: "<< Base :: len << "\n";
    cout << "The original array:";
}
```

Файл Cl1.h

```
#ifndef CL1_H
#define CL1_H

#include "Base.h"
class Cl1 : virtual public Base{
public:
    void Fill();
}
```

```
};
#endif
```

Файл Cl2.cpp

```
#include "Cl2.h"
#include <iostream>
#include <iomanip>
#include <algorithm>
using namespace std;

void Cl2 :: Sort(){
    cout<<"\n";
    for (int i=0;i<Base::len-1;i++)
        for (int j=i+1;j<Base::len;j++)
            if (Base::arr[i]>Base::arr[j]) swap(Base::arr[i],
Base::arr[j]);
    cout << "An ordered array:";
}
void Cl2 :: Print(){
    for (int i=0;i<Base::len;i++)
        cout<<setw(5)<<Base::arr[i];
}
```

Файл Cl2.h

```
#ifndef CL2_H
#define CL2_H

#include "Base.h"
class Cl2 : virtual public Base{
public:
    void Sort();
    void Print();
};

#endif
```

Файл Cl3.cpp

```

#include "Cl3.h"
#include <iostream>
#include <iomanip>
using namespace std;

void Cl3 :: Method(){
    Cl1 :: Fill();

    Cl2 :: Print();

    Cl2 :: Sort();

    Cl2 :: Print();
}

```

Файл Cl3.h

```

#ifndef CL3_H
#define CL3_H

#include "Cl1.h"
#include "Cl2.h"
class Cl3 : public Cl1, public Cl2{
public:
    void Method();
};

#endif

```

Файл main.cpp

```

#include "Base.h"
#include "Cl1.h"
#include "Cl2.h"
#include "Cl3.h"
#include <iostream>
using namespace std;

int main(){
    Cl3 obj;
    obj.Method();

    return 0;
}

```

Тестирование

Результат тестирования программы представлен в следующей таблице.

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
6 9 1 3 4 2 1	Array dimension: 6 The original array: 9 1 3 4 2 1 An ordered array: 1 1 2 3 4 9	Array dimension: 6 The original array: 9 1 3 4 2 1 An ordered array: 1 1 2 3 4 9
5 5 4 3 2 1	Array dimension: 5 The original array: 5 4 3 2 1 An ordered array: 1 2 3 4 5	Array dimension: 5 The original array: 5 4 3 2 1 An ordered array: 1 2 3 4 5

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ (ИСТОЧНИКОВ)

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.
3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avrrora.ru/student/files/methodichescoe_posobie_dlya_laboratornyh_rabot_3.pdf (дата обращения 05.05.2021).
4. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avrrora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).
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).