

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	6
1.1 Описание входных данных.....	7
1.2 Описание выходных данных.....	7
2 МЕТОД РЕШЕНИЯ.....	8
3 ОПИСАНИЕ АЛГОРИТМОВ.....	11
3.1 Алгоритм конструктора класса class1.....	11
3.2 Алгоритм метода Print класса class1.....	11
3.3 Алгоритм конструктора класса class2.....	12
3.4 Алгоритм метода Print класса class2.....	12
3.5 Алгоритм конструктора класса class3.....	12
3.6 Алгоритм метода Print класса class3.....	13
3.7 Алгоритм конструктора класса class4.....	13
3.8 Алгоритм метода Print класса class4.....	14
3.9 Алгоритм функции main.....	14
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	16
5 КОД ПРОГРАММЫ.....	18
5.1 Файл class1.cpp.....	18
5.2 Файл class1.h.....	18
5.3 Файл class2.cpp.....	19
5.4 Файл class2.h.....	19
5.5 Файл class3.cpp.....	20
5.6 Файл class3.h.....	20
5.7 Файл class4.cpp.....	21
5.8 Файл class4.h.....	21
5.9 Файл main.cpp.....	22
6 ТЕСТИРОВАНИЕ.....	23

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	24
---------------------------------------	----

1 ПОСТАНОВКА ЗАДАЧИ

Иерархия наследования

Описать четыре класса которые последовательно наследуют друг друга, последовательными номерами классов 1,2,3,4.

Реализовать программу, в которой использовать единственный указатель на объект базового класса (номер класса 1).

Наследственность реализовать так, что можно было вызывать методы, принадлежащие объекту конкретного класса, только через объект данного класса.

В закрытом разделе каждого класса определены два свойства: строкового типа для наименования объекта и целого типа для значения определенного целочисленного выражения.

Описание каждого класса содержит один параметризованный конструктор с строковым и целочисленным параметром.

В реализации каждого конструктора объекта определяются значения закрытых свойств:

- Наименование объекта по шаблону: «значение строкового параметра»_«номер класса»;
- Целочисленного свойства значением выражения возведения в степень номера класса целочисленного значения параметра конструктора.

Еще в описании каждого класса определен метод с одинаковым наименованием для всех классов, реализующий вывод значений закрытых свойств класса.

В основной функции реализовать алгоритм:

1. Вводится идентификатор и натуральное число от 2 до 10.
2. Создать объект класса 4, используя параметризованный конструктор,

которому в качестве аргументов передаются введенный идентификатор и натуральное число.

3. Построчно, для всех объектов согласно наследственности, от объекта базового (класс 1) до производного объекта (класса 4) вывести наименование объекта класса и значение целочисленного свойства.

1.1 Описание входных данных

Первая строка:

«идентификатор» «натуральное число»

Пример ввода:

Object 2

1.2 Описание выходных данных

Построчно (четыре строки):

«идентификатор»_«номер класса» «значение целочисленного свойства»

Разделитель - 1 пробел.

Пример вывода:

Object_1 2
Object_2 4
Object_3 8
Object_4 16

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект obj класса class 4 предназначен для демонстрация иерархии наследования;
- cin - объект стандартного потока ввода;
- cout - объект стандартного потока вывода.

Класс class1:

- свойства/поля:
 - поле хранения названия объекта:
 - наименование — name;
 - тип — string;
 - модификатор доступа — private;
 - поле хранения значения объекта:
 - наименование — n;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - метод class1 — конструктор, устанавливает имя и целочисленное значение объекту;
 - метод Print — выводит имя объекта и его значение.

Класс class2:

- свойства/поля:
 - поле хранения названия объекта:
 - наименование — name;
 - тип — string;
 - модификатор доступа — private;

- о поле хранения значения объекта:
 - наименование — `n`;
 - тип — `int`;
 - модификатор доступа — `private`;
- функционал:
 - о метод `class2` — конструктор, устанавливает имя и целочисленное значение объекту;
 - о метод `Print` — выводит имя объекта и его значение.

Класс `class3`:

- свойства/поля:
 - о поле хранения названия объекта:
 - наименование — `name`;
 - тип — `string`;
 - модификатор доступа — `private`;
 - о поле хранения значения объекта:
 - наименование — `n`;
 - тип — `int`;
 - модификатор доступа — `private`;
- функционал:
 - о метод `class3` — конструктор, устанавливает имя и целочисленное значение объекту;
 - о метод `Print` — выводит имя объекта и его значение.

Класс `class4`:

- свойства/поля:
 - о поле хранения названия объекта:
 - наименование — `name`;
 - тип — `string`;

- модификатор доступа — private;
- о поле хранения значения объекта:
 - наименование — n;
 - тип — int;
 - модификатор доступа — private;
- функционал:
 - о метод class4 — конструктор, устанавливает имя и целочисленное значение объекту;
 - о метод Print — выводит имя объекта и его значение.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	class1			родительский класс, содержит основные поля и методы	
2	class2			дочерний класс класса class1	
		class1	private		1
3	class3			дочерний класс класса class2	
		class2	private		2
4	class4			Дочерний класс класса class3	
		class3	private		3

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 конструктора класса *class1*

Функционал: устанавливает имя и целочисленное значение объекту.

Параметры: string name, int n.

Алгоритм конструктора представлен в таблице 2.

Таблица 2 – Алгоритм конструктора класса *class1*

№	Предикат	Действия	№ перехода
1		присвоить скрытому свойству name значение параметра name и дописать "_1"	2
2		присвоить скрытому свойству n значение параметра n	∅

3.2 Алгоритм метода Print класса *class1*

Функционал: выводит имя объекта и его значение.

Параметры: none.

Возвращаемое значение: -.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода Print класса *class1*

№	Предикат	Действия	№ перехода
1		вывод значений скрытых свойств name и n через пробел	∅

3.3 Алгоритм конструктора класса class2

Функционал: устанавливает имя и целочисленное значение объекту.

Параметры: string name, int n.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса class2

№	Предикат	Действия	№ перехода
1		присвоить скрытому свойству name значение параметра name и дописать "_2"	2
2		присвоить скрытому свойству n значение параметра n*n	∅

3.4 Алгоритм метода Print класса class2

Функционал: выводит имя объекта и его значение.

Параметры: none.

Возвращаемое значение: -.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода Print класса class2

№	Предикат	Действия	№ перехода
1		вывод значений скрытых свойств name и n через пробел	∅

3.5 Алгоритм конструктора класса class3

Функционал: конструктор, устанавливает имя и целочисленное значение объекту.

Параметры: string name, int n.

Алгоритм конструктора представлен в таблице 6.

Таблица 6 – Алгоритм конструктора класса class3

№	Предикат	Действия	№ перехода
1		присвоить скрытому свойству name значение параметра name и дописать "_3"	2
2		присвоить скрытому свойству n значение параметра n*n*n	∅

3.6 Алгоритм метода Print класса class3

Функционал: выводит имя объекта и его значение.

Параметры: none.

Возвращаемое значение: -.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода Print класса class3

№	Предикат	Действия	№ перехода
1		вывод значений скрытых свойств name и n через пробел	∅

3.7 Алгоритм конструктора класса class4

Функционал: конструктор, устанавливает имя и целочисленное значение объекту.

Параметры: string name, int n.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса class4

№	Предикат	Действия	№ перехода
1		присвоить скрытому свойству name значение параметра name и дописать "_4"	2
2		присвоить скрытому свойству n значение параметра n*n*n*n	∅

3.8 Алгоритм метода Print класса class4

Функционал: выводит имя объекта и его значение.

Параметры: none.

Возвращаемое значение: -.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода Print класса class4

№	Предикат	Действия	№ перехода
1		вывод значений скрытых свойств name и n через пробел	Ø

3.9 Алгоритм функции main

Функционал: запуск программы.

Параметры: none.

Возвращаемое значение: код ошибки int.

Алгоритм функции представлен в таблице 10.

Таблица 10 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление переменной типа string	2
2		объявление переменной n типа int	3
3		ввод значения переменной name	4
4		ввод значения переменной n	5
5		инициализация указателя obj на объект класса class1 адресом нового объекта класса class4 с помощью оператора функции new с передачей конструктору параметров name, n с приведением к указателю на объект класса class1	6
6		вызов метода Print() объекта obj	7
7		вывод переноса на новую строку	8

№	Предикат	Действия	№ перехода
8		вызов метода Print() объекта obj, приведенный к указателю на объект класса class2	9
9		вывод переноса на новую строку	10
10		вызов метода Print() объектом obj, приведенный к указателю на объект класса class3	11
11		вывод переноса на новую строку	12
12		вызов метода Print() объекта obj приведенный к указателю на объект класса class4	13
13		освобождение памяти, выделенной под объект obj с помощью оператора функции delete	14
14		возврат значения 0	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

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

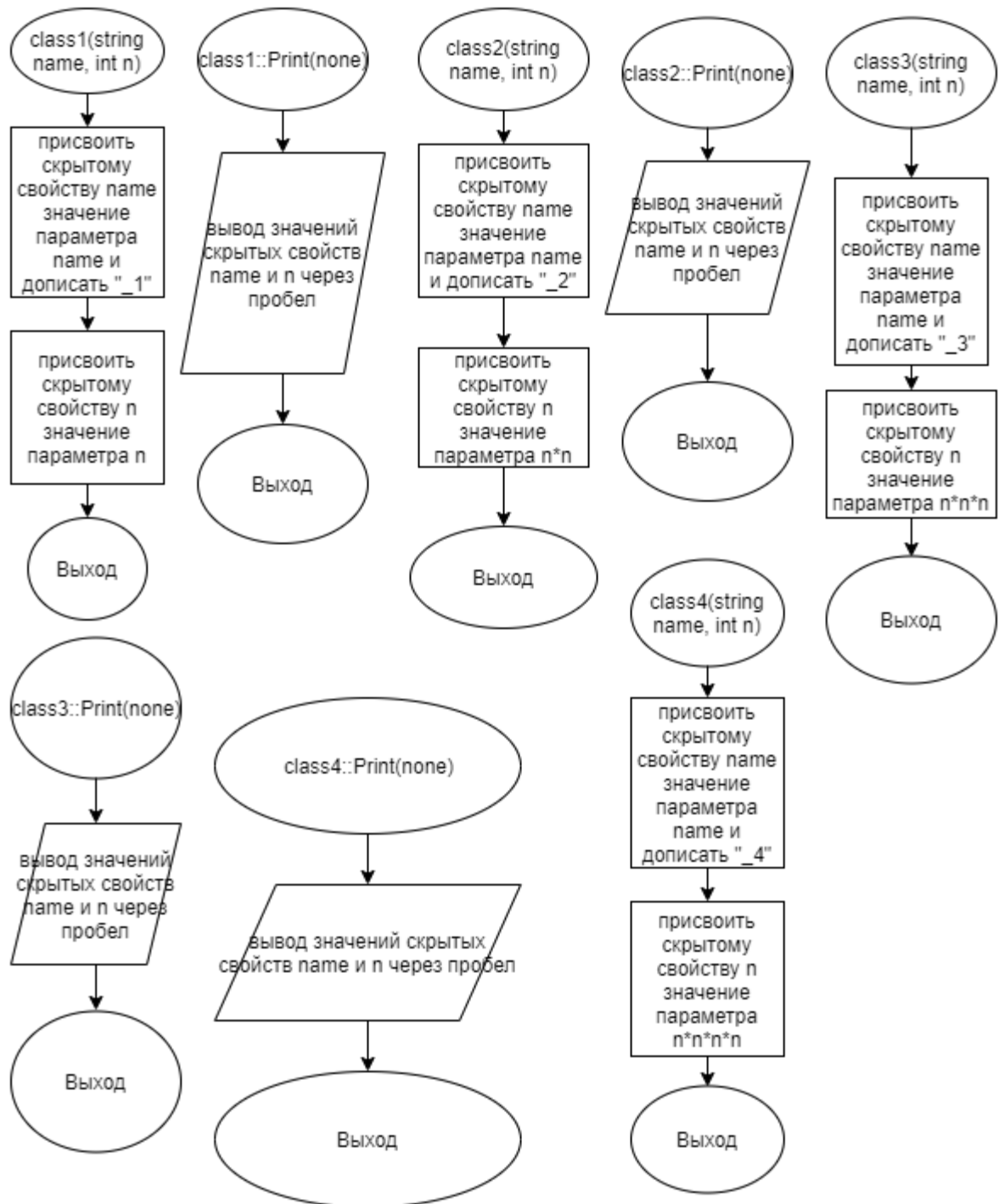


Рисунок 1 – Блок-схема алгоритма

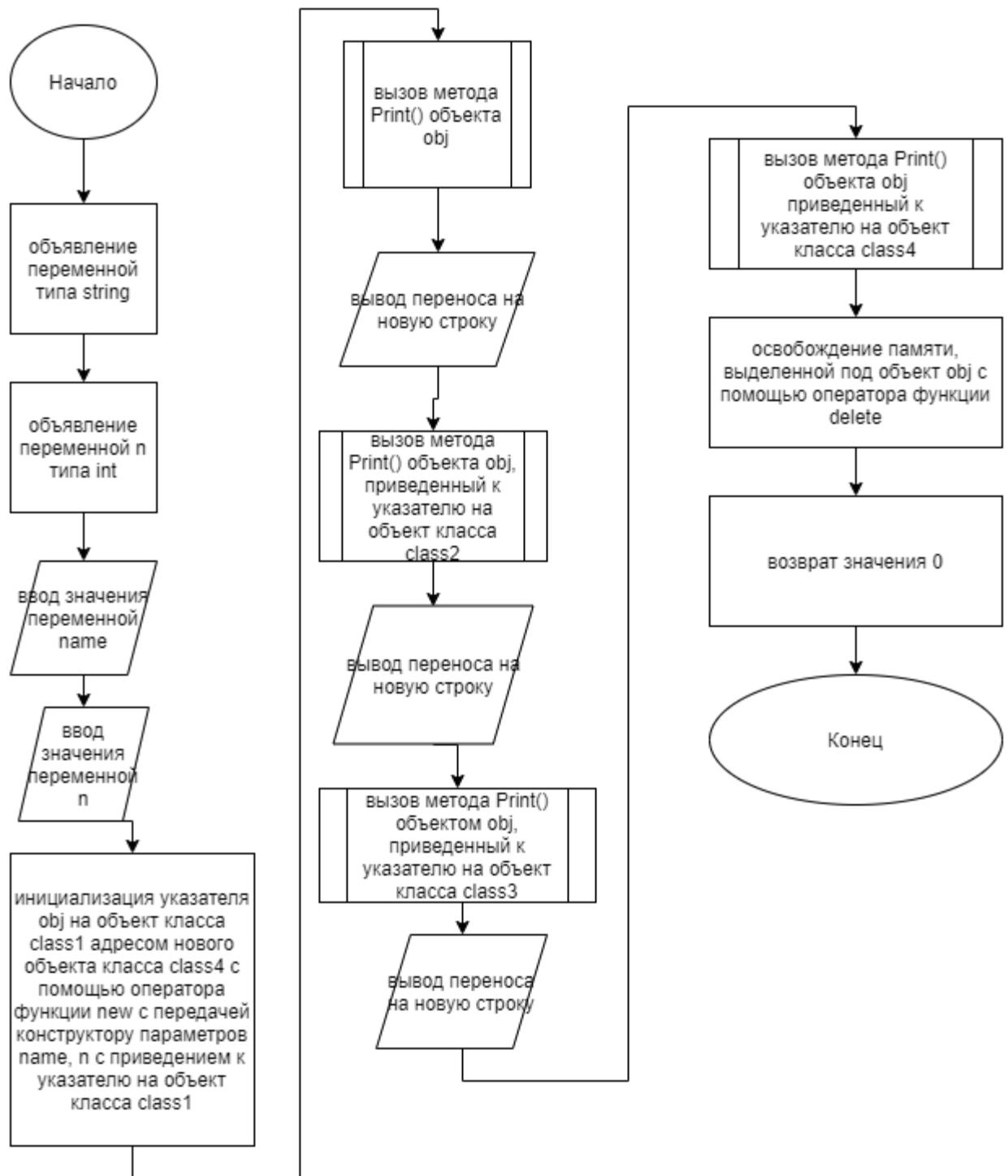


Рисунок 2 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл class1.cpp

Листинг 1 – class1.cpp

```
#include "class1.h"
#include <iostream>

class1::class1(string name, int n)
{
    this->name=name+"_1";
    this->n=n;
}
void class1::Print()
{
    cout<<name<<" ";
    cout<<n;
}
```

5.2 Файл class1.h

Листинг 2 – class1.h

```
#ifndef __CLASS1__H
#define __CLASS1__H
#include <iostream>

using namespace std;

class class1
{
private:
    string name;
    int n;
public:
    class1(string name, int n);
    void Print();
};
```



```
#endif
```

5.3 Файл class2.cpp

Листинг 3 – class2.cpp

```
#include "class2.h"
#include "class1.h"
#include <iostream>

class2::class2(string name, int n):class1::class1(name, n)
{
    this->name=name+"_2";
    this->n=n*n;
}
void class2::Print()
{
    cout<<name<<" ";
    cout<<n;
}
```

5.4 Файл class2.h

Листинг 4 – class2.h

```
#ifndef __CLASS2__H
#define __CLASS2__H
#include "class1.h"
#include <iostream>

using namespace std;

class class2:private class1
{
    private:
        string name;
        int n;
    public:
        class2(string name, int n);
        void Print();
};

#endif
```

5.5 Файл class3.cpp

Листинг 5 – class3.cpp

```
#include "class3.h"
#include "class2.h"

using namespace std;

class3::class3(string name, int n):class2::class2(name, n)
{
    this->name=name+"_3";
    this->n=n*n*n;
}
void class3::Print()
{
    cout<<name<<" ";
    cout<<n;
}
```

5.6 Файл class3.h

Листинг 6 – class3.h

```
#ifndef __CLASS3__H
#define __CLASS3__H
#include "class2.h"
#include <iostream>

using namespace std;

class class3:private class2
{
    private:
        string name;
        int n;
    public:
        class3(string name, int n);
        void Print();
};

#endif
```

5.7 Файл class4.cpp

Листинг 7 – class4.cpp

```
#include "class4.h"
#include "class3.h"

using namespace std;

class4::class4(string name, int n):class3::class3(name, n)
{
    this->name=name+"_4";
    this->n=n*n*n*n;
}
void class4::Print()
{
    cout<<name<<" ";
    cout<<n;
}
```

5.8 Файл class4.h

Листинг 8 – class4.h

```
#ifndef __CLASS4__H
#define __CLASS4__H
#include "class3.h"
#include <iostream>

using namespace std;

class class4:private class3
{
    private:
        string name;
        int n;
    public:
        class4(string name, int n);
        void Print();
};

#endif
```

5.9 Файл main.cpp

Листинг 9 – main.cpp

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include "class1.h"
#include "class2.h"
#include "class3.h"
#include "class4.h"

int main()
{
    string name;
    int n;
    cin>>name;
    cin>>n;
    class1* obj=(class1*)new class4(name, n);
    obj->Print();
    cout<<endl;
    ((class2*)obj)->Print();
    cout<<endl;
    ((class3*)obj)->Print();
    cout<<endl;
    ((class4*)obj)->Print();
    delete obj;
    return(0);
}
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 11.

Таблица 11 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
Object 2	Object_1 2 Object_2 4 Object_3 8 Object_4 16	Object_1 2 Object_2 4 Object_3 8 Object_4 16
Object 69	Object_1 69 Object_2 4761 Object_3 328509 Object_4 22667121	Object_1 69 Object_2 4761 Object_3 328509 Object_4 22667121

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).