Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ

Разработать систему, которая демонстрирует работу (использование) конструктора копии.

Первоначально, в процессе конструирования, система состоит из одного объекта (исходного) и из одной функции. Далее, ее работа управляется посредством команд. Команды вводятся с стандартного потока данных. По команде создаются копии объектов от исходного или от других копии. Копия так же создается при передаче объекта в функцию по значению.

Спроектировать объект, с свойствами в закрытом доступе:

- строкового типа, для хранения наименования объекта;
- целого типа, для хранения номера копии объекта;
- целого типа, для хранения размерности массива;
- указатель на объект целого типа, для хранения адреса динамически созданного целочисленного массива;
- свойство, для хранения целочисленного значения, который доступен всем копиям (инструкцию static не использовать);
- еще из дополнительных свойств, для реализации требований, заданных в постановке задачи.

С параметризированным конструктором. У конструктора есть параметр строкового типа. Параметр передает (содержит) значение наименования объекта. В реализации метода фиксируется имя объекта. Свойству номера копии присваивается значение нуль. Общедоступному свойству тоже присваивается нуль. Есть фрагмент, в котором вводится значение размерности целочисленного массива, динамический создается целочисленный массив согласно значению размерности и вводятся значения элементов массива.

С конструктором копии. В конструкторе копии из переданному в качестве

параметра объекту в текущем:

- копируется наименование объекта;
- определяется номер новой копии и присваивается соответствующему свойству;
- определяется значение свойства с общим доступом;
- копируется размерность массива:
- динамический создается целочисленный массив согласно значению размерности. Элементам массива присваиваются значения элементов исходного массива увеличенные на значение номера копии текущего объекта;
- выводиться наименование объекта и номер копии.

Имеется функционал (методы) в открытом доступе:

- с одним целочисленным параметром, значением которого редактируется значение свойства с общим доступом;
- возвращает значение номера копии объекта;
- выводит состояние объекта; наименование объекта; номер копии; значение свойства с общим доступом; значения элементов массива;
- деструктор, который сообщает объект с каким именем и с каким номер копии уничтожается, освобождается память, выделенная для целочисленного массива.

В составе системы определена функция с одним параметром. По этому параметру передается объект по значению. В функции для объекта, переданного по параметру, вызывается метод вывода состояния объекта.

Система работает по следующим командам:

```
copy «целое число, номер копии»
```

По данной команде ищется объект с заданным номером копии, создается новый объект посредством конструктора копии, в котором в качестве аргумента

передается найденный объект. Выводиться состояние нового объекта.

```
function «целое число, номер копии»
```

По данной команде ищется объект с заданным номером копии, вызывается функция и в качестве аргумента передается найденный объект.

```
state «целое число, номер копии»
```

По данной команде ищется объект с заданным номером копии и выводится его состояние.

```
shared «целое число, номер копии» «целое число, значение свойства с общим доступом»
```

По данной команде ищется объект с заданным номером копии. Вводиться значение свойства с общим доступом. Для найденного объекта вызывается метод редактирования значения свойства с общим доступом.

```
end
```

По данной команде система завершает работу.

Если не удается найти объект с данным номер копии, то выдается соответствующее сообщение.

Алгоритм конструирования и отработки системы:

1. Объявляются строковые переменные, для хранения наименования объекта и значения команд системы.
2. Объявляются целочисленные переменные для хранения значений: номера копии объекта и значения свойства с общим доступом.
3. Могут быть другие объявления.
4. Вводится значение наименования объекта.
5. Выводится значение наименования объекта.
6. Создается объект посредством параметризированного конструктора.
7. Для созданного объекта вызывается метод вывода состояния объекта.
8. Могут быть другие операторы.

9. Организуется цикл по командам.

1.1 Описание входных данных

Первая строка:

```
«строка, наименование объекта»
```

Вторая строка:

```
«целое число, размерность массива»
```

Третья строка:

```
«целое число» «целое число» . . . «целое число»
```

Значения элементов массива.

Начиная с четвертой строки, построчно, вводятся команды системы.

```
команда «целое число, номер копии» [«целое число, значение свойства с общим доступом»]
```

Пример ввода

```
Document
5
1 2 3 4 5
copy 0
copy 1
shared 8 99
copy 1
shared 2 77
copy 0
function 1
state 5
state 1
end
```

1.2 Описание выходных данных

Каждый вывод производится с новой строки.

Вывод наименования объекта:

```
Object name: «наименование объекта»
```

Вывод состояния объекта занимает две строки. Первая строка:

```
Object name: «наименование объекта»      Copy: «номер копии»      Shared: «значение свойства с общим доступом»
```

Вторая строка:

```
Array:  «целое число»  «целое число» . . . «целое число»
```

Сообщение конструктора копии имеет следующий шаблон:

```
Construcrot copy Object name: «наименование объекта»    Copy: «номер копии»
```

Если объект с заданным номером копии не найден, то выводиться сообщение:

```
A copy of object number «номер копии» was not found.
```

Сообщение деструктора копии имеет следующий шаблон:

```
Destructor Object name: «наименование объекта»    Copy: «номер копии»
```

Пример вывода

```
Object name: Document
Object name: Document    Copy: 0    Shared: 0
Array:  1  2  3  4  5
Construcrot copy Object name: Document    Copy: 1
Object name: Document    Copy: 1    Shared: 0
Array:  2  3  4  5  6
Construcrot copy Object name: Document    Copy: 2
Object name: Document    Copy: 2    Shared: 0
Array:  4  5  6  7  8
A copy of object number 8 was not found.
Construcrot copy Object name: Document    Copy: 3
Object name: Document    Copy: 3    Shared: 0
Array:  5  6  7  8  9
Construcrot copy Object name: Document    Copy: 4
Object name: Document    Copy: 4    Shared: 77
Array:  5  6  7  8  9
Construcrot copy Object name: Document    Copy: 5
Object name: Document    Copy: 5    Shared: 77
Array:  7  8  9  10  11
Destructor Object name: Document    Copy: 5
A copy of object number 5 was not found.
Object name: Document    Copy: 1    Shared: 77
Array:  2  3  4  5  6
```

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект original класса Object предназначен для ;
- функция execute для вызова метода вывода состояния объекта, переданного по параметру;
- функция get_object_by_number для получения объекта по его номеру копии;
- функция main для реализации основного алгоритма программы;
- библиотека iostream;
- библиотека string;
- библиотека vector;
- оператор new;
- объекты стандартного потока вывода cin и cout;
- условный оператор if...else;
- оператор цикла со счётчиком for;
- оператор цикла с предусловием while.

Класс Object:

- свойства/поля:
    - поле для хранения целочисленного значения, которое доступно всем копиям:
        - наименование — value;
        - тип — целочисленное значение;
        - модификатор доступа — public;
    - поле для хранения наименования объекта:
        - наименование — name;
        - тип — строка;

        - модификатор доступа — private;
    - поле для хранения номера копии объекта:
        - наименование — number;
        - тип — целочисленное значение;
        - модификатор доступа — private;
    - поле для хранения размерности массива:
        - наименование — array_size;
        - тип — целочисленное значение;
        - модификатор доступа — private;
    - поле для хранения адреса динамически созданного целочисленного массива:
        - наименование — array;
        - тип — указатель на объект целого типа;
        - модификатор доступа — private;
- функционал:
    - метод set_value — редактирование значения свойства с общим доступом;
    - метод get_number — возврат значения номера копии объекта;
    - метод print_info — вывод состояния объекта;
    - метод Object — параметризированный конструктор;
    - метод Object — конструктор копии;
    - метод ~Object — деструктор.

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 set_value класса Object

Функционал: редактирование значения свойства с общим доступом.

Параметры: целочисленное значение new_value.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 1.

Таблица 1 – Алгоритм метода set_value класса Object

№	Предикат	Действия	№ перехода
1		Присвоение значения параметра полю, хранящему общедоступное значение	∅

3.2 Алгоритм метода get_number класса Object

Функционал: возврат значения номера копии объекта.

Параметры: нет.

Возвращаемое значение: целочисленное значение.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода get_number класса Object

№	Предикат	Действия	№ перехода
1		Возврат значения номера копии объекта	∅

3.3 Алгоритм метода print_info класса Object

Функционал: вывод состояния объекта.

Параметры: нет.

Возвращаемое значение: нет.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода print_info класса Object

№	Предикат	Действия	№ перехода
1		Вывод значений полей, содержащих имя объекта, номер копии и общедоступное значение	2
2		Инициализация целочисленного счётчика значением, равным 0	3
3	Значение счётчика МЕНЬШЕ значения размерности массива	Вывод значения элемента целочисленного массива с индексом, равным значению целочисленного счётчика	4
			∅
4		Увеличение значения счётчика на 1	3

3.4 Алгоритм конструктора класса Object

Функционал: параметризированный конструктор.

Параметры: строка name - имя создаваемого объекта.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса Object

№	Предикат	Действия	№ перехода
1		Присвоение значения параметра полю, хранящему имя текущего объекта	2
2		Присвоение полю, хранящему номер копии	3

№	Предикат	Действия	№ перехода
		объекта значения, равного нулю	
3		Присвоение общедоступному полю значения, равного нулю	4
4		Ввод значения размерности массива	5
5		Создание динамического массива заданной размерности	6
6		Инициализация целочисленного счётчика значением, равным нулю	7
7	Значение счётчика МЕНЬШЕ значения размерности массива	Ввод значения элемента массива с индексом, равным значению счётчика	8
			∅
8		Увеличение значения счётчика на 1	7

3.5 Алгоритм деструктора класса Object

Функционал: деструктор.

Параметры: нет.

Алгоритм деструктора представлен в таблице 5.

Таблица 5 – Алгоритм деструктора класса Object

№	Предикат	Действия	№ перехода
1		Вывод имени объекта и его номера копии	∅

3.6 Алгоритм функции execute

Функционал: вызов метода вывода состояния объекта, переданного по параметру.

Параметры: объект object класса Object - объект, для которого нужно

вызвать метод вывода состояния.

Возвращаемое значение: нет.

Алгоритм функции представлен в таблице 6.

Таблица 6 – Алгоритм функции execute

№	Предикат	Действия	№ перехода
1		Вызов метода print_info для объекта, переданного в параметре	∅

3.7 Алгоритм функции main

Функционал: реализациия основного алгоритма программы.

Параметры: нет.

Возвращаемое значение: целочисленное значение.

Алгоритм функции представлен в таблице 7.

Таблица 7 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявляются строковые переменные, для хранения наименования объекта и значения команд системы	2
2		Объявляются целочисленные переменные для хранения значений: номера копии объекта и значения свойства с общим доступом	3
3		Объявление контейнера для объектов системы типа vector с указателями на объект класса Object	4
4		Ввод значения наименования объекта.	5
5		Вывод значения наименования объекта.	6
6		Создание объекта класса Object посредством параметризированного конструктора с передачей значения наименования объекта в качестве	7

№	Предикат	Действия	№ перехода
		параметра	
7		Вызов метода print_info для созданного объекта	8
8		Добавление созданного объекта в контейнер для объектов системы	9
9		Объявление двух целочисленных переменных для хранения параметров вводимых команд	10
10		Ввод команды	11
11	Значение команды РАВНО "copy"	Ввод значения параметра команды	12
			16
12		Инициализация объекта target_object класса Object значением, полученным в результате вызова функции get_object_by_number с передачей контейнера объектов и значения параметра команды в качестве параметров	13
13	Значение target_object РАВНО значению, равному нулевому указателю		9
		Создание объекта класса Object посредством параметризированного конструктора с передачей значения указателя target_object, и инкрементированного значения номера копии в качестве параметров	14
14		Добавление созданного объекта в контейнер для объектов системы	15
15		Вызов метода print_info для созданного объекта	9
16	Значение команды РАВНО "function"	Ввод значения параметра команды	17
			20

№	Предикат	Действия	№ перехода
17		Инициализация объекта target_object класса Object значением, полученным в результате вызова функции get_object_by_number с передачей контейнера объектов и значения параметра команды в качестве параметров	18
18	Значение target_object РАВНО значению, равному нулевому указателю		9
		Создание объекта класса Object посредством параметризированного конструктора с передачей значения указателя target_object, и инкрементированного значения номера копии в качестве параметров	19
19		Вызов функции execute с передачей созданного объекта в качестве параметра	9
20	Значение команды РАВНО "state"	Ввод значения параметра команды	21
			23
21		Инициализация объекта target_object класса Object значением, полученным в результате вызова функции get_object_by_number с передачей контейнера объектов и значения параметра команды в качестве параметров	22
22	Значение target_object РАВНО значению, равному нулевому указателю		9
		Вызов метода print_info для элемента контейнера для объектов системы с индексом, равным значению параметра команды	9
23	Значение команды РАВНО	Ввод значений параметров команды	24

№	Предикат	Действия	№ перехода
	"shared"		
			26
24		Инициализация объекта target_object класса Object значением, полученным в результате вызова функции get_object_by_number с передачей контейнера объектов и значения параметра команды в качестве параметров	25
25	Значение target_object РАВНО значению, равному нулевому указателю		9
		Вызов метода set_value для элемента контейнера для объектов системы с индексом, равным значению первого параметра команды, с передачей значения второго параметра команды в качестве параметра	9
26	Значение команды РАВНО "end"		∅
			9

3.8 Алгоритм конструктора класса Object

Функционал: конструктор копии.

Параметры: константная ссылка на объект класса Object original - исходный объект, копия которого будет создаваться; целочисленное значение copy_number - номер копии.

Алгоритм конструктора представлен в таблице 8.

Таблица 8 – Алгоритм конструктора класса Object

№	Предикат	Действия	№ перехода
1		Присвоение значения наименования исходного объекта полю, содержащему наименование текущего объекта	2
2		Присвоение значения параметра, содержащего номер копии объекта полю, содержащему номер копии объекта	3
3		Присвоение значения поля (указателя), содержащего адрес общедоступного значения полю (указателю), содержащему адрес общедоступного значения	4
4		Присвоение значения размерности массива исходного объекта полю, содержащему значение размерности массива текущего объекта	5
5		Создание динамического массива заданной размерности	6
6		Инициализация целочисленного счётчика значением, равным нулю	7
7	Значение счётчика МЕНЬШЕ значения размерности массива	Присвоение элементу целочисленного массива текущего объекта с индексом, равным значению счётчика, значения, равного значению элемента целочисленного массива исходного объекта с индексом, равным значению счётчика, увеличенного на значение номера копии объекта	8
			9
8		Увеличение значения счётчика на 1	9
9		Вывод наименования объекта и его номера копии	∅

3.9 Алгоритм функции get_object_by_number

Функционал: функция для получения объекта из контейнера по его номеру копии.

Параметры: контейнер типа vector с указателями на объект класса Object - контейнер, в котором будет осуществляться поиск; целочисленное значение number - номер копии искомого объекта.

Возвращаемое значение: указатель на объект класса Object.

Алгоритм функции представлен в таблице 9.

Таблица 9 – Алгоритм функции get_object_by_number

№	Предикат	Действия	№ перехода
1		Инициализация целочисленного счётчика значением, равным нулю	2
2	Значение счётчика МЕНЬШЕ значения размерности контейнера объектов		3
			4
3	Результат вызова метода get_number для элемента контейнера с индексом, равным значению счетчика РАВЕН значению параметра, хранящего номер искомой копии	Возврат значения элемента контейнера с индексом, равным значению счетчика	∅
		Увеличение значения счётчика на 1	2
4		Вывод сообщения о том, что не удается найти объект с данным номером копии	5
5		Возврат значения, равного нулевому указателю	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-9.

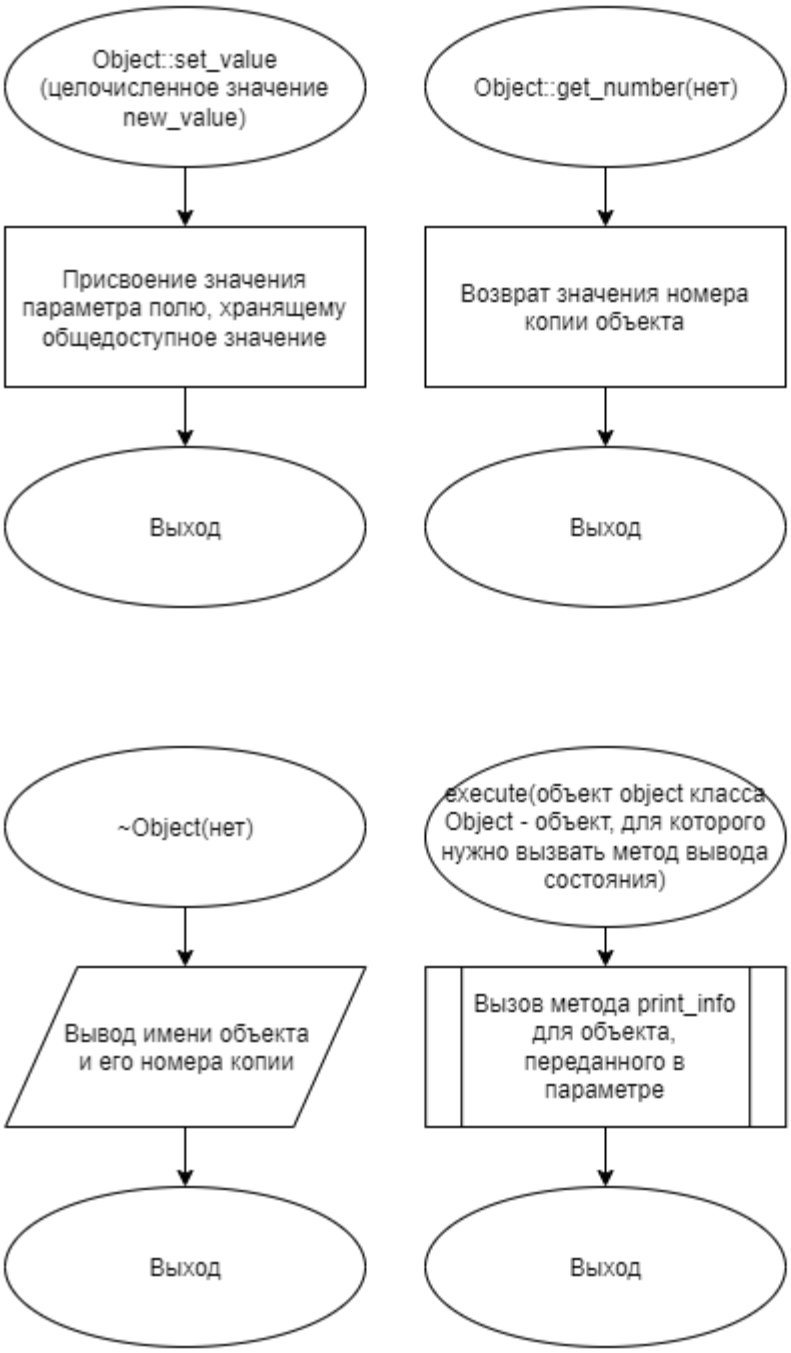

Рисунок 1 – Блок-схема алгоритма

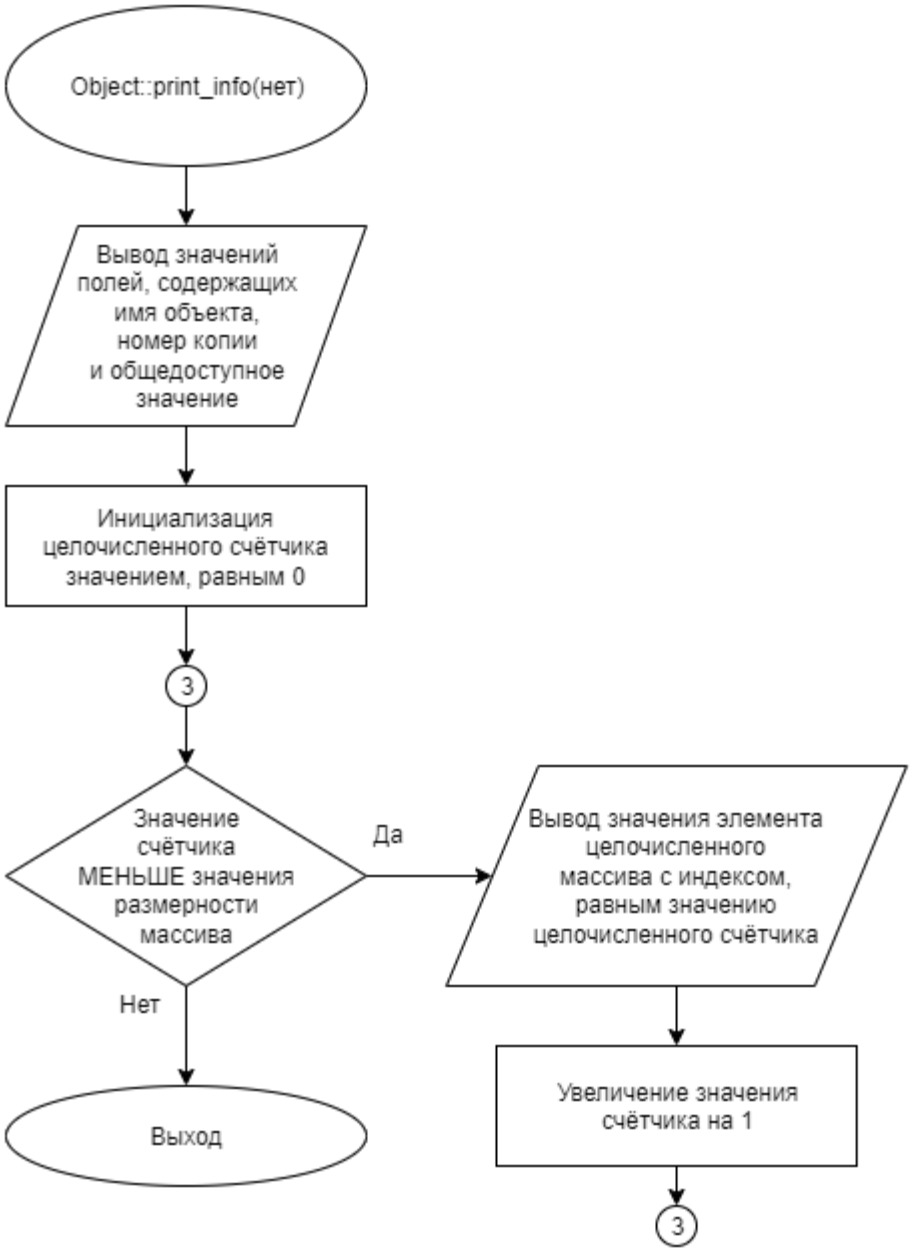

Рисунок 2 – Блок-схема алгоритма

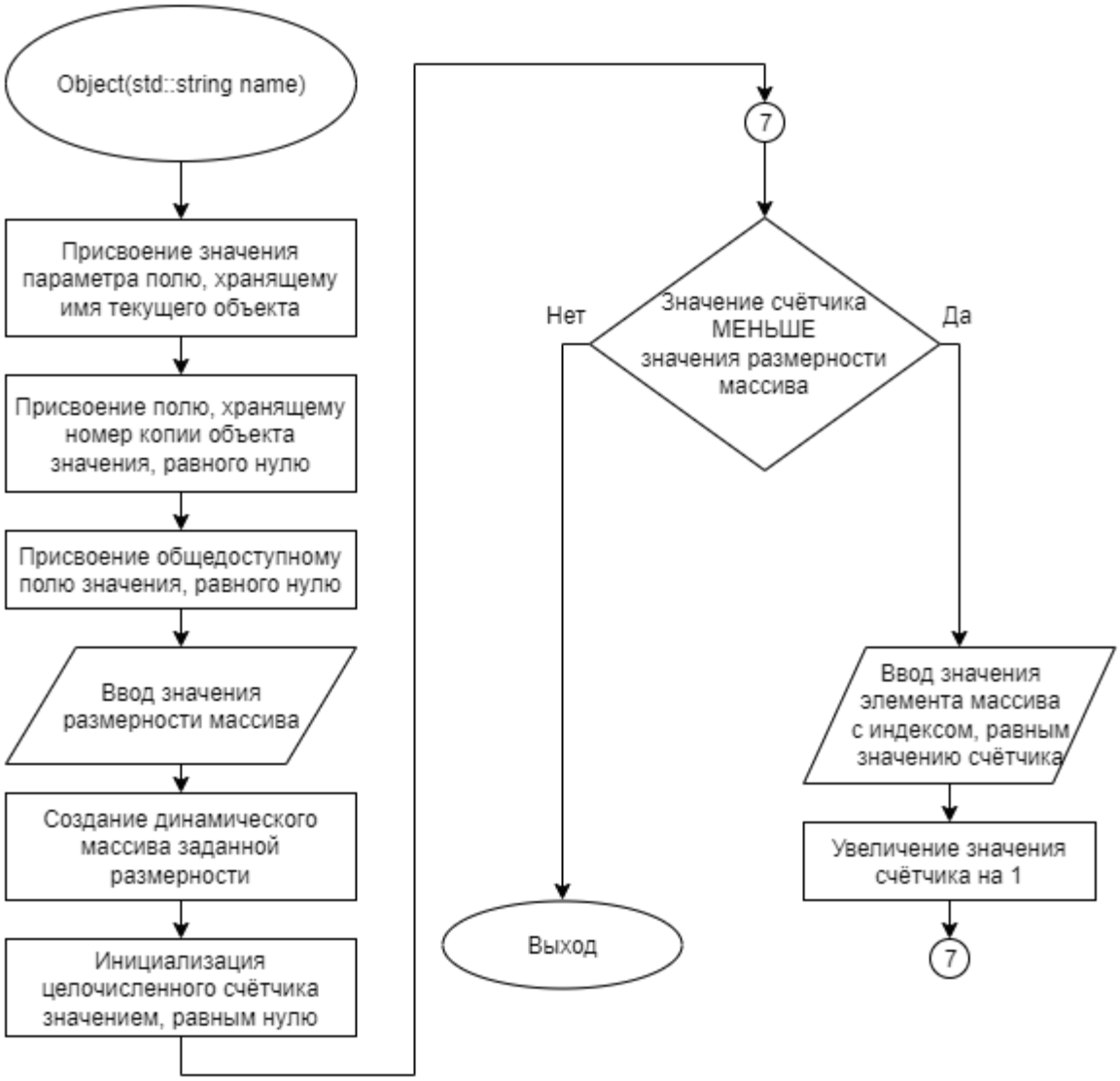

Рисунок 3 – Блок-схема алгоритма

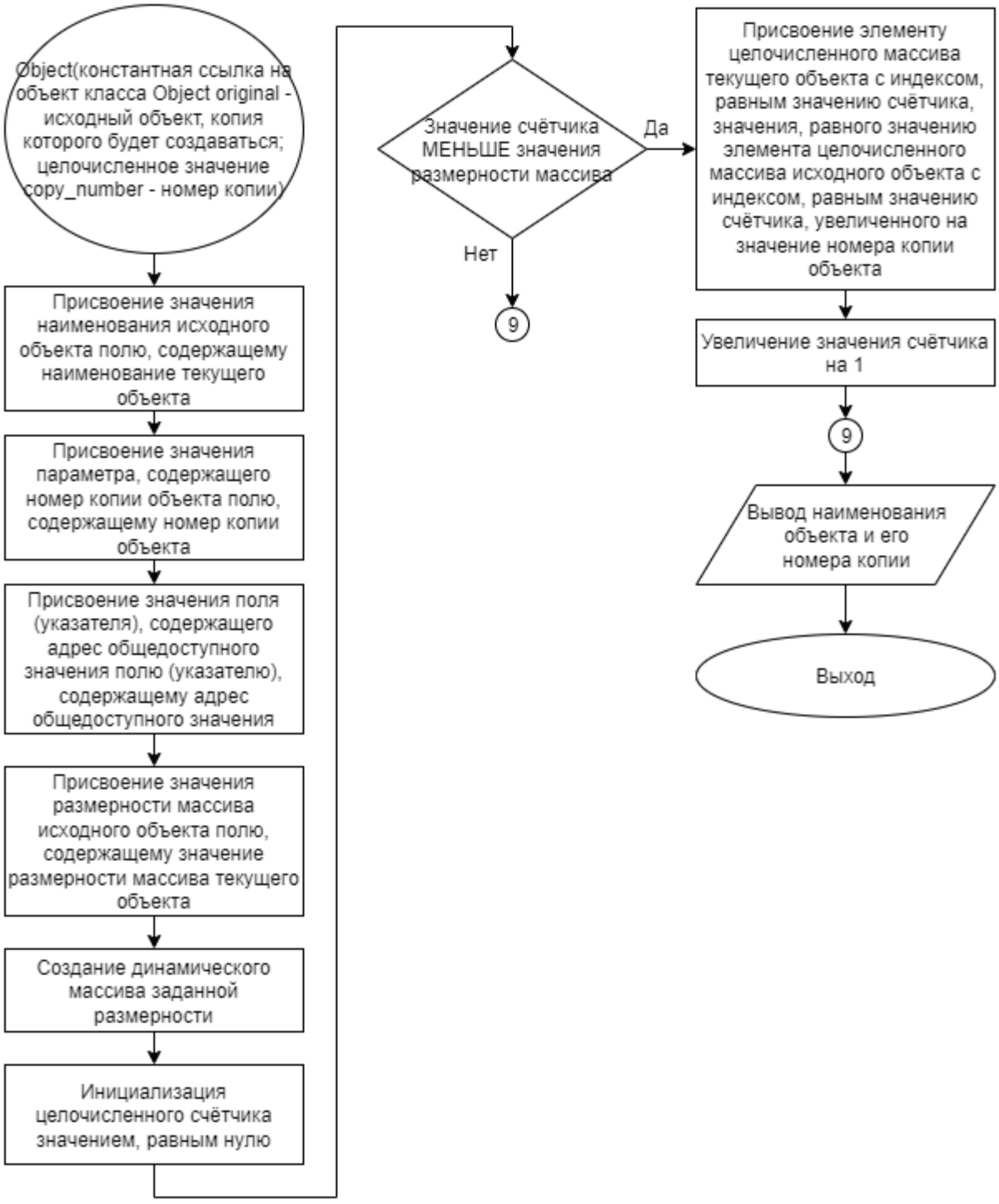

Рисунок 4 – Блок-схема алгоритма

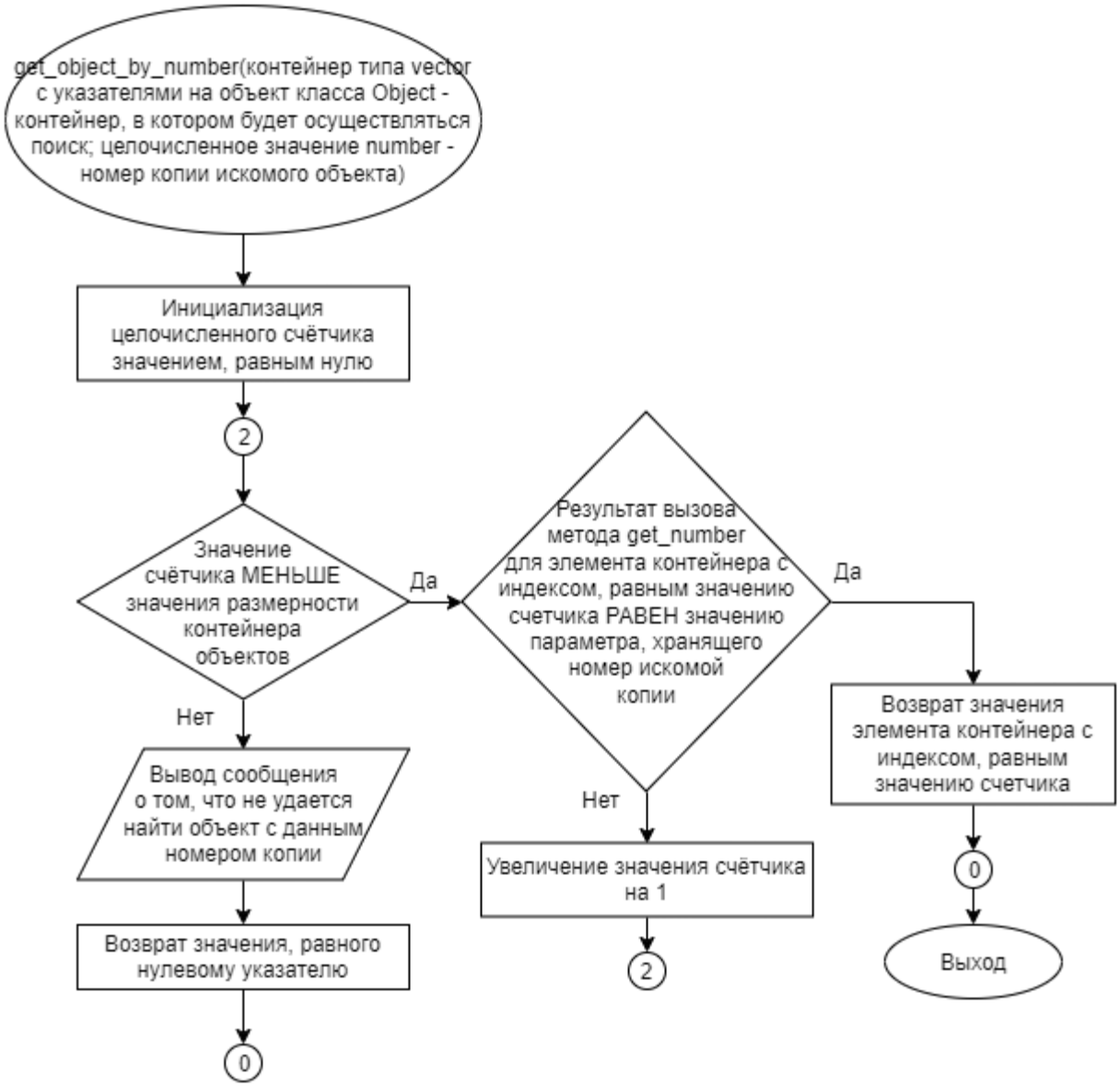

Рисунок 5 – Блок-схема алгоритма

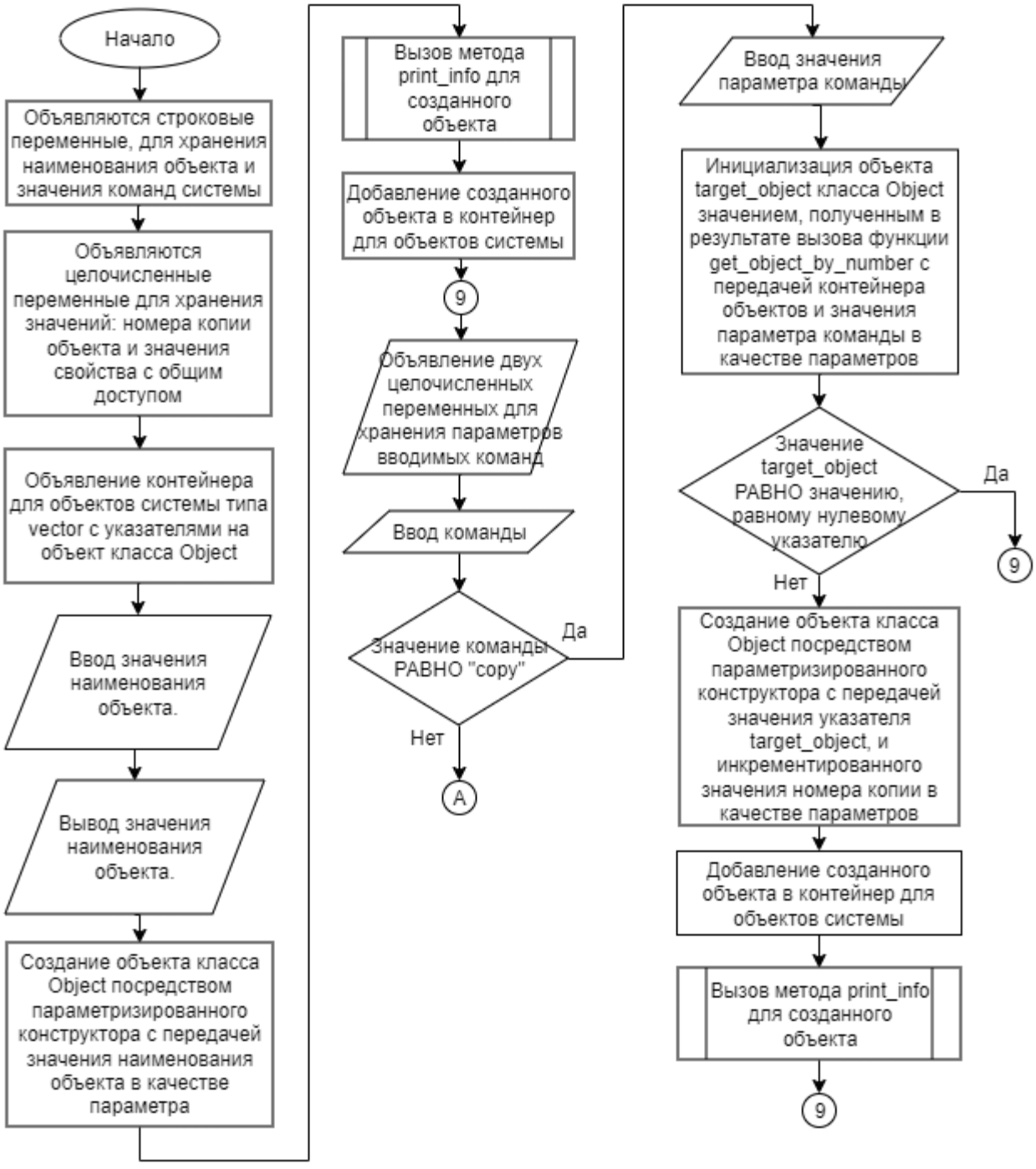

Рисунок 6 – Блок-схема алгоритма

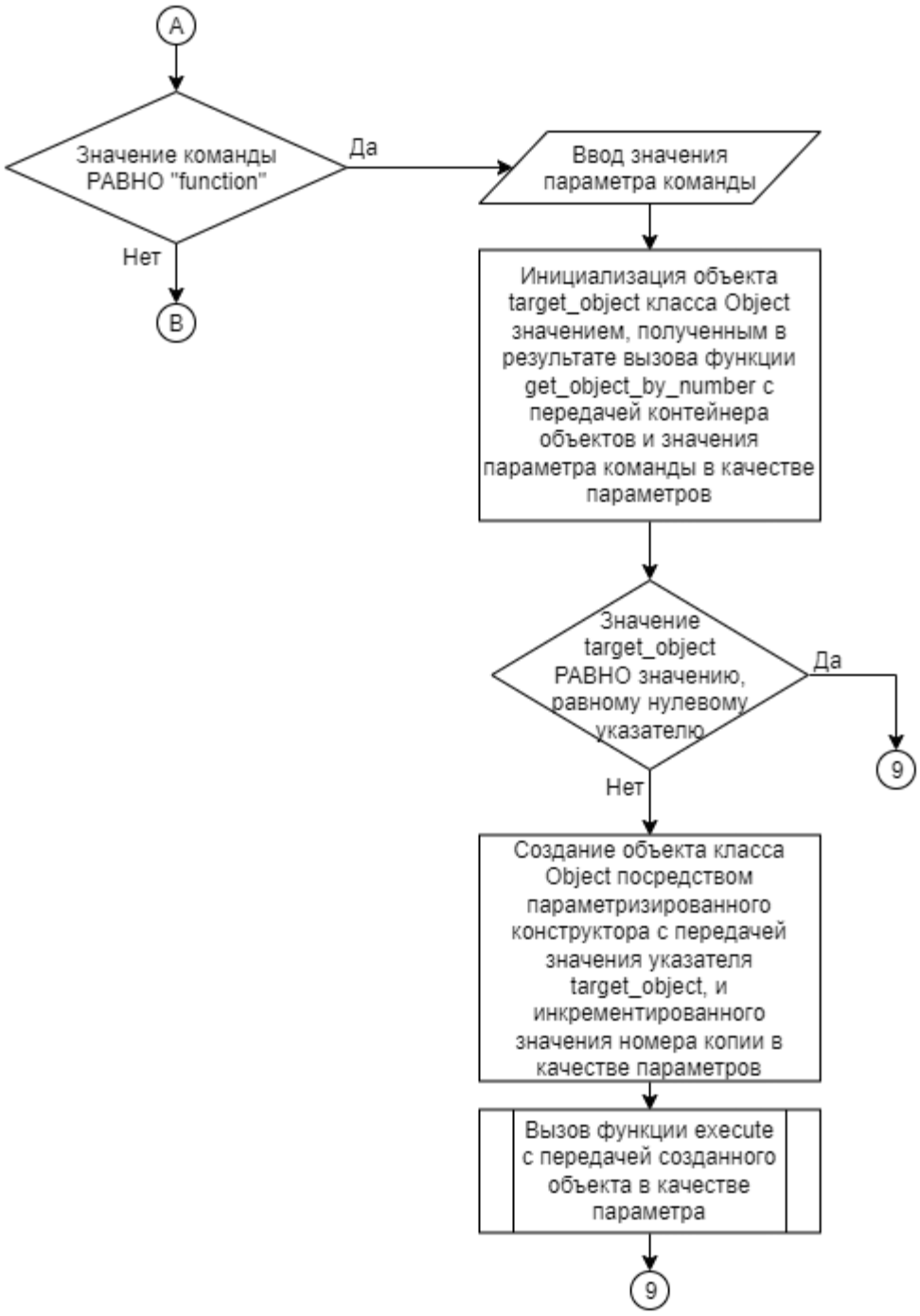

Рисунок 7 – Блок-схема алгоритма

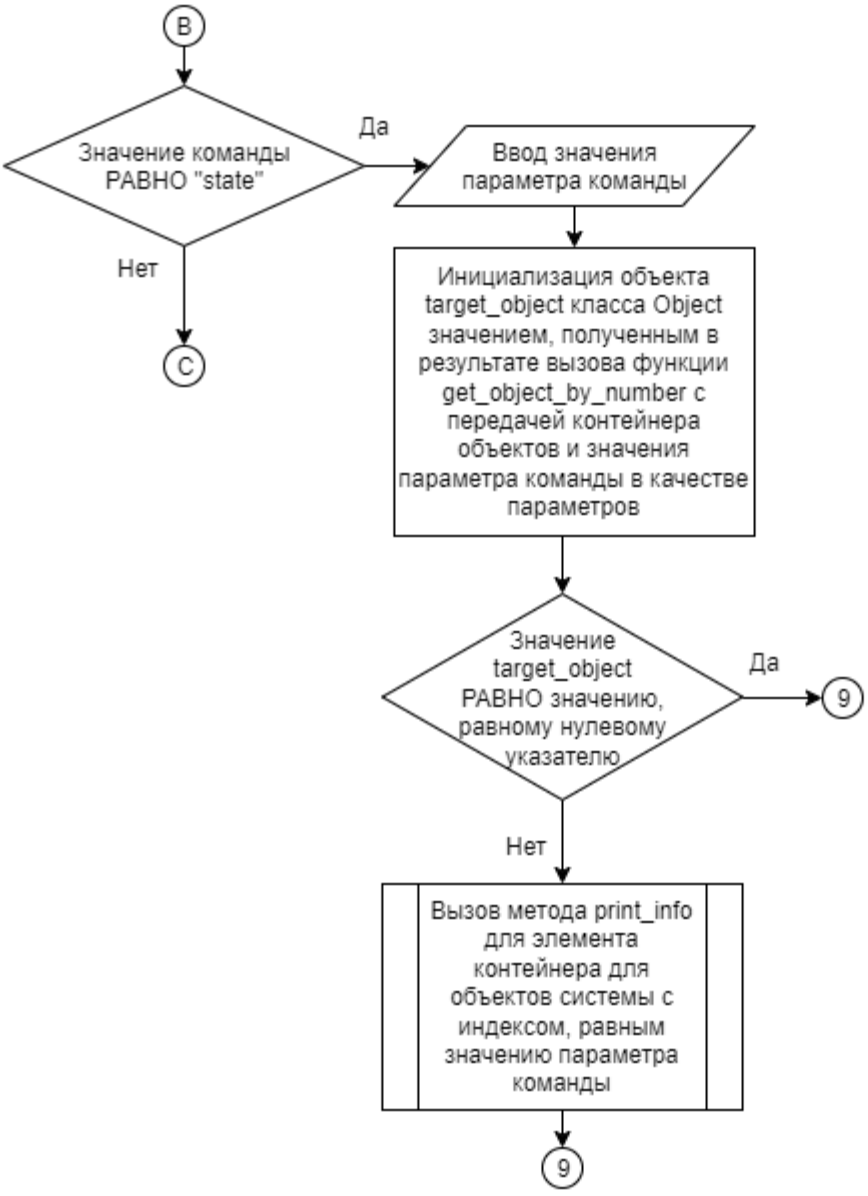

Рисунок 8 – Блок-схема алгоритма

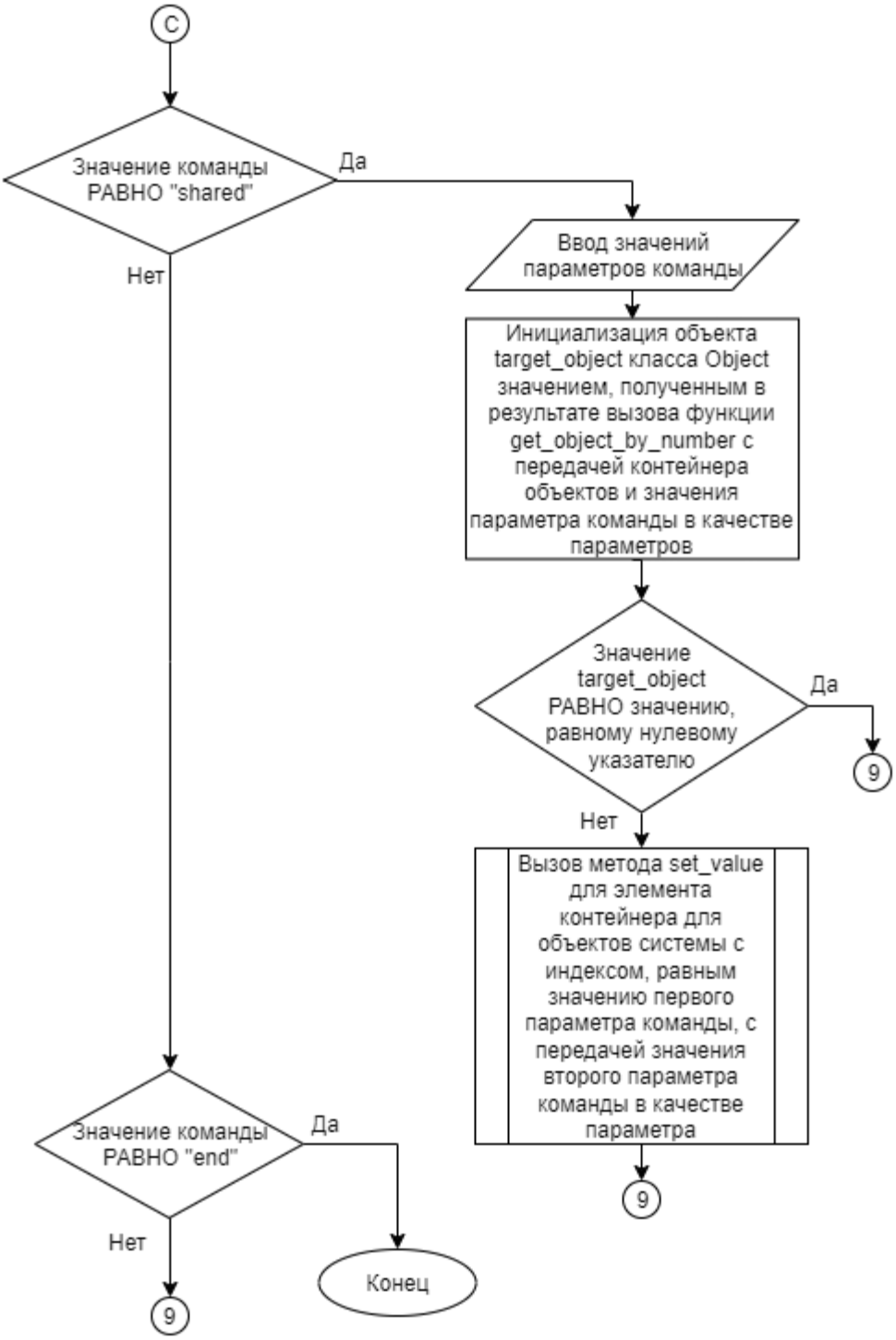

Рисунок 9 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include "Object.h"
#include <vector>

void execute(Object object)
{
   object.print_info();
}

Object* get_object_by_number(std::vector<Object*>& container, int number)
{
   for(int i = 0; i < container.size(); i++)
   {
      if(container[i]->get_number() == number)
      {
         return container[i];
      }
   }
   std::cout << "A copy of object number " << number << " was not found." <<
std::endl;
   return nullptr;
}

int main()
{
   std::string name = "", command = "";
   int copy_number = 0, value = 0;

   std::vector<Object*> objects;

   std::cin >> name;
   std::cout << "Object name: " << name << std::endl;
   Object* original = new Object(name);
   original->print_info();

   objects.push_back(original);

   while(true)
   {
      int arg_1 = 0, arg_2 = 0;
```

```
            std::cin >> command;

            if(command == "copy")
            {
               std::cin >> arg_1;
               Object* target_object = get_object_by_number(objects, arg_1);

               if(target_object == nullptr) continue;
               Object* obj = new Object(*target_object, ++copy_number);
               objects.push_back(obj);
               obj->print_info();
            }
            else if(command == "function")
            {
               std::cin >> arg_1;
               Object* target_object = get_object_by_number(objects, arg_1);

               if(target_object == nullptr) continue;
               Object* obj = new Object(*target_object, ++copy_number);
               execute(*obj);
            }
            else if(command == "state")
            {
               std::cin >> arg_1;
               Object* target_object = get_object_by_number(objects, arg_1);

               if(target_object == nullptr) continue;
               target_object->print_info();
            }
            else if(command == "shared")
            {
               std::cin >> arg_1;
               std::cin >> arg_2;
               Object* target_object = get_object_by_number(objects, arg_1);

               if(target_object == nullptr) continue;
               target_object->set_value(arg_2);
            }
            else if(command == "end")
            {
               break;
            }
         }

         return 0;
      }
```

5.2 Файл Object.cpp

Листинг 2 – Object.cpp

```
#include "Object.h"


void Object::set_value(int new_value)
{
   *value = new_value;
}

int Object::get_number()
{
   return number;
}

void Object::print_info()
{
   std::cout << "Object name: " << name << "     Copy: " << number << "
Shared: " << *value << std::endl;
   std::cout << "Array:";

   for(int i = 0; i < array_size; i++)
   {
      std::cout << "  " << array[i];
   }
   std::cout << std::endl;
}

Object::Object(std::string name)
{
   this->name = name;
   number = 0;
   value = new int(0);

   std::cin >> array_size;
   array = new int[array_size];
   for(int i = 0; i < array_size; i++)
   {
      std::cin >> array[i];
   }
}

Object::Object(const Object& original, int copy_number)
{
   name = original.name;
   number = copy_number;
   value = original.value;
   array_size = original.array_size;

   array = new int[array_size];
   for(int i = 0; i < array_size; i++)
   {
```

```
      array[i] = original.array[i] + number;
   }

   std::cout << "Construcrot copy Object name: " << name << "    Copy: " <<
number << std::endl;
}

Object::~Object()
{
   std::cout << "Destructor Object name: " << name << "    Copy: " << number
<< std::endl;
}
```

5.3 Файл Object.h

Листинг 3 – Object.h

```
#ifndef __OBJECT__H
#define __OBJECT__H

#include <iostream>
#include <string>


class Object
{
public:
   int* value;

   void set_value(int new_value);
   int get_number();
   void print_info();

   Object(std::string name);
   Object(const Object& original, int copy_number);
   ~Object();

private:
   std::string name;
   int number;
   int array_size;
   int* array;

};

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 10.

Таблица 10 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
Document 5 1 2 3 4 5 copy 0 copy 1 shared 8 99 copy 1 shared 2 77 copy 0 function 1 state 5 state 1 end	Object name: Document Object name: Document Copy: 0 Shared: 0 Array: 1 2 3 4 5 Construcrot copy Object name: Document Copy: 1 Object name: Document Copy: 1 Shared: 0 Array: 2 3 4 5 6 Construcrot copy Object name: Document Copy: 2 Object name: Document Copy: 2 Shared: 0 Array: 4 5 6 7 8 A copy of object number 8 was not found. Construcrot copy Object name: Document Copy: 3 Object name: Document Copy: 3 Shared: 0 Array: 5 6 7 8 9 Construcrot copy Object name: Document Copy: 4 Object name: Document Copy: 4 Shared: 77 Array: 5 6 7 8 9 Construcrot copy Object name: Document Copy: 5 Object name:	Object name: Document Object name: Document Copy: 0 Shared: 0 Array: 1 2 3 4 5 Construcrot copy Object name: Document Copy: 1 Object name: Document Copy: 1 Shared: 0 Array: 2 3 4 5 6 Construcrot copy Object name: Document Copy: 2 Object name: Document Copy: 2 Shared: 0 Array: 4 5 6 7 8 A copy of object number 8 was not found. Construcrot copy Object name: Document Copy: 3 Object name: Document Copy: 3 Shared: 0 Array: 5 6 7 8 9 Construcrot copy Object name: Document Copy: 4 Object name: Document Copy: 4 Shared: 77 Array: 5 6 7 8 9 Construcrot copy Object name: Document Copy: 5 Object name:

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
	Document Copy: 5 Shared: 77 Array: 7 8 9 10 11 Destructor Object name: Document Copy: 5 A copy of object number 5 was not found. Object name: Document Copy: 1 Shared: 77 Array: 2 3 4 5 6	Document Copy: 5 Shared: 77 Array: 7 8 9 10 11 Destructor Object name: Document Copy: 5 A copy of object number 5 was not found. Object name: Document Copy: 1 Shared: 77 Array: 2 3 4 5 6
Plan 7 1 2 3 4 5 6 7 copy 0 copy 2 copy 1 state 8 99 copy 1 shared 0 177 function 3 copy 0 function 1 state 5 state 1 end	Object name: Plan Object name: Plan Copy: 0 Shared: 0 Array: 1 2 3 4 5 6 7 Construcrot copy Object name: Plan Copy: 1 Object name: Plan Copy: 1 Shared: 0 Array: 2 3 4 5 6 7 8 A copy of object number 2 was not found. Construcrot copy Object name: Plan Copy: 2 Object name: Plan Copy: 2 Shared: 0 Array: 4 5 6 7 8 9 10 A copy of object number 8 was not found. Construcrot copy Object name: Plan Copy: 3 Object name: Plan Copy: 3 Shared: 0 Array: 5 6 7 8 9 10 11 Construcrot copy Object name: Plan Copy: 4 Object name: Plan Copy: 4 Shared:	Object name: Plan Object name: Plan Copy: 0 Shared: 0 Array: 1 2 3 4 5 6 7 Construcrot copy Object name: Plan Copy: 1 Object name: Plan Copy: 1 Shared: 0 Array: 2 3 4 5 6 7 8 A copy of object number 2 was not found. Construcrot copy Object name: Plan Copy: 2 Object name: Plan Copy: 2 Shared: 0 Array: 4 5 6 7 8 9 10 A copy of object number 8 was not found. Construcrot copy Object name: Plan Copy: 3 Object name: Plan Copy: 3 Shared: 0 Array: 5 6 7 8 9 10 11 Construcrot copy Object name: Plan Copy: 4 Object name: Plan Copy: 4 Shared:

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
	177 Array: 9 10 11 12 13 14 15 Destructor Object name: Plan Copy: 4 Construcrot copy Object name: Plan Copy: 5 Object name: Plan Copy: 5 Shared: 177 Array: 6 7 8 9 10 11 12 Construcrot copy Object name: Plan Copy: 6 Object name: Plan Copy: 6 Shared: 177 Array: 8 9 10 11 12 13 14 Destructor Object name: Plan Copy: 6 Object name: Plan Copy: 5 Shared: 177 Array: 6 7 8 9 10 11 12 Object name: Plan Copy: 1 Shared: 177 Array: 2 3 4 5 6 7 8	177 Array: 9 10 11 12 13 14 15 Destructor Object name: Plan Copy: 4 Construcrot copy Object name: Plan Copy: 5 Object name: Plan Copy: 5 Shared: 177 Array: 6 7 8 9 10 11 12 Construcrot copy Object name: Plan Copy: 6 Object name: Plan Copy: 6 Shared: 177 Array: 8 9 10 11 12 13 14 Destructor Object name: Plan Copy: 6 Object name: Plan Copy: 5 Shared: 177 Array: 6 7 8 9 10 11 12 Object name: Plan Copy: 1 Shared: 177 Array: 2 3 4 5 6 7 8

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).