Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ

Сконструировать систему, которая демонстрирует отличие проектирования объекта посредством объединения от проектирования посредством описания класса. Также демонстрирует особенности обработки и интерпретации значений свойств объекта.

Спроектировать объект с:

- одним целочисленным свойством типа int открытого доступа;
- целочисленным массивом типа char открытого доступа, для хранения четырех символов;
- одним целочисленным свойством типа short закрытого доступа;
- функционалом открытого доступа для редактирования значения свойства закрытого доступа;
- функционалом открытого доступа для получения значения этого свойства.

Создать два описания этого объекта посредством объединения и класса. В описании посредством объединения ключевое слово public не использовать. В описании посредством класса ключевое слово private не использовать.

В описании объединения, в методе реализующей функционал редактирования значения свойства закрытого доступа, исходное значение (значение параметра) присваивается значению этого свойства.Символы, которые используются входных данных принадлежат латинскому алфавиту. Значения целых чисел во входных данных представлены в шестнадцатеричном формате. При этом, значение пары шестнадцатеричных цифр соответствует коду буквы латинского алфавита. Например: целочисленное значение 19533 в шестнадцатеричном формате 0x4C4D. Запись соответствует буквам LM. Другие целочисленные данные считаются ошибочным.

Алгоритм отработки системы:

1. Создает объект с использованием описания объединения obj_un.
2. Создает объект с использованием описания класса obj_cl.
3. Вводит значения четырех символов и размещает в элементах массива объекта obj_un и obj_cl.
4. Вводит целочисленное значения, для открытого свойства.
5. Если значение равно нулю, система завершает работу.
6. Если значение из допустимого множества, то переход к пункту 8.
7. Иначе, вывод сообщения об ошибочных данных и переход к пункту 4.
8. Вводит целочисленное значение, для закрытого свойства.
9. Если значение из допустимого множества, то переход к пункту 11.
10. Иначе, вывод сообщения об ошибочных данных и переход к пункту 4.
11. Введенное значение присваивает к открытому свойству объекта obj_un и obj_cl.
12. Введенное значение присваивает к закрытому свойству объекта obj_un и obj_cl.
13. Выводит значения содержимого элементов массива объекта obj_un и obj_cl.
14. Переход к пункту 4.

1.1 Описание входных данных

Первая строка:

«Символ латинского алфавита»«Символ латинского алфавита»«Символ латинского алфавита»«Символ латинского алфавита»

Вторая и последующие строки:

«Целое число в шестнадцатеричном формате» «Целое число в шестнадцатеричном формате»

Последняя строка:

```
0x00
```

0x00

Пример ввода:

```
ABCD
0x41424344 0x4C4D
0x41424344 0x3C4D
0x41324344 0x4C4D
0x00
```

1.2 Описание выходных данных

Каждый вывод с новой строки.

Вывод содержимого массиве объекта obj_un:

```
«символ»«символ»«символ»«символ»
```

Вывод содержимого массиве объекта obj_st:

```
«символ»«символ»«символ»«символ»
```

Вывод сообщения об ошибке в данных закрытого типа:

```
Data of the short type is incorrect: «число в шестнадцатеричном формате»
```

Вывод сообщения об ошибке в данных открытого типа:

```
Data of the int type is incorrect: «число в шестнадцатеричном формате»
```

Пример вывода:

```
MLBA
ABCD
Data of the short type is incorrect: 3c4d
Data of the int type is incorrect: 41324344
```

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- объект un класса obj_un предназначен для реализации структуры данных через объединение;
- объект cl класса obj_cl предназначен для реализации структуры данных через класс;
- функция main для работы основного алгоритма программы;
- функция checkString для проверки, принадлежит ли символ алфавиту латинских букв;
- функция printArray для вывода массива символов;
- библиотека iostream;
- оператор стандартного потока ввода данных - cin;
- оператор стандартного потока вывода данных - cout;
- оператор перехода на следующую строку - endl;
- операторы логического ветвления - if/else;
- оператор цикла с предусловием - while;
- оператор цикла с параметром - for;
- оператор прерывания цикла - break;
- оператор прерывания итерации цикла - continue.

Класс obj_un:

- свойства/поля:
    - поле представитель типа int:
        - наименование — open_digit;
        - тип — int;
        - модификатор доступа — public;
    - поле представитель типа char:

        - наименование — symbol_pack[4];
        - тип — char;
        - модификатор доступа — public;
    - поле представитель типа short:
        - наименование — hidden_digit;
        - тип — short;
        - модификатор доступа — private;
- функционал:
    - метод setDigit — определение значения поля hidden_digit;
    - метод getDigit — доступ к полю hidden_digit.

Класс obj_cl:

- свойства/поля:
    - поле представитель типа int:
        - наименование — open_digit;
        - тип — int;
        - модификатор доступа — public;
    - поле представитель типа char:
        - наименование — symbol_pack[4];
        - тип — char;
        - модификатор доступа — public;
    - поле представитель типа short:
        - наименование — hidden_digit;
        - тип — short;
        - модификатор доступа — private;
- функционал:
    - метод setDigit — определение значения поля hidden_digit;
    - метод getDigit — доступ к полю hidden_digit.

Таблица 1 – Иерархия наследования классов

№	Имя класса	Классы-наследники	Модификатор доступа при наследовании	Описание	Номер
1	obj_un				
2	obj_cl				

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 setDigit класса obj_un

Функционал: определение значения поля hidden_digit.

Параметры: short n - определяет значение поля hidden_digit.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода setDigit класса obj_un

№	Предикат	Действия	№ перехода
1		определение поля hidden_digit	∅

3.2 Алгоритм метода getDigit класса obj_un

Функционал: доступ к полю hidden_digit.

Параметры: нет.

Возвращаемое значение: short - значение поля hidden_digit.

Алгоритм метода представлен в таблице 3.

Таблица 3 – Алгоритм метода getDigit класса obj_un

№	Предикат	Действия	№ перехода
1		вывод значения поля hidden_digit	∅

3.3 Алгоритм метода setDigit класса obj_cl

Функционал: определение значения поля hidden_digit.

Параметры: short n - определяет значение поля hidden_digit.

Возвращаемое значение: void.

Алгоритм метода представлен в таблице 4.

Таблица 4 – Алгоритм метода setDigit класса obj_cl

№	Предикат	Действия	№ перехода
1		определение значения поля hidden_digit значением n	∅

3.4 Алгоритм метода getDigit класса obj_cl

Функционал: доступ к полю hidden_digit.

Параметры: нет.

Возвращаемое значение: short - значение поля hidden_digit.

Алгоритм метода представлен в таблице 5.

Таблица 5 – Алгоритм метода getDigit класса obj_cl

№	Предикат	Действия	№ перехода
1		вывод значения поля hidden_digit	∅

3.5 Алгоритм функции checkString

Функционал: проверка, что каждый байт числа digit соответствует латинской букве.

Параметры: int digit - целое число, интерпретируемое как последовательность байтов.

Возвращаемое значение: bool - индикатор корректности введенной строки.

Алгоритм функции представлен в таблице 6.

Таблица 6 – Алгоритм функции checkString

№	Предикат	Действия	№ перехода
1		объявление переменной symbol типа int	2
2		инициализация переменной flag значением true	3
3	в строке еще существуют буквы?	определение переменной symbol значением деления с остатком переменной digit на 256	4
			6
4		определение переменной digit значением целочисленного деления ее на 256	5
5	символ удовлетворяет латинским буквам?		3
		определение переменной flag значением false	3
6		вывод значения переменной flag	∅

3.6 Алгоритм функции printArray

Функционал: вывод содержимого массива.

Параметры: char[] arr - массив, символы которого будут выведены.

Возвращаемое значение: void.

Алгоритм функции представлен в таблице 7.

Таблица 7 – Алгоритм функции printArray

№	Предикат	Действия	№ перехода
1		инициализация переменной i типа int со значением 0	2
2	i < 4?	вывод элемента массива arr по индексу i	3
		вывод перехода на новую строку	∅
3		инкремент переменной i на 1	2

3.7 Алгоритм функции main

Функционал: главная функция программы.

Параметры: нет.

Возвращаемое значение: целое, идентификатор работоспособности программы.

Алгоритм функции представлен в таблице 8.

Таблица 8 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		объявление переменной chars типа string	2
2		ввод в переменную chars с консоли	3
3		объявление объекта un класса obj_un	4
4		объявление объекта cl класса obj_cl	5
5		инициализация переменной i типа int со значением 0	6
6	i < 4?	заполнение поля symbol_pack объекта cl элементом массива chars по индексу i	7
			9
7		заполнение поля symbol_pack объекта un элементом массива chars по индексу i	8
8		инкремент переменной i на значение 1	6
9	бесконечный цикл?		∅
		объявление переменных intDigit, shortDigit	10
10		определение переменной intDigit значением 1	11
11		ввод шестнадцатеричного представления целого числа в переменную intDigit	12
12	intDigit == 0?		∅
			13
13	intDigit можно представить в		14

№	Предикат	Действия	№ перехода
	виде латинских букв?		
		вывод строки "Data of the int type is incorrect: " и значения intDigit	9
14		ввод шестнадцатеричного представления целого числа в переменную shortDigit	15
15	shortDigit == 0?		∅
			16
16	shortDigit можно представить в виде латинских букв?		17
		Вывод строки "Data of the short type is incorrect: " и значения shortDigit	9
17		определение поля open_digit объекта un значением intDigit	18
18		определение поля open_digit объекта cl значением intDigit	19
19		вызов метода setDigit объекта un аргументом shortDigit	20
20		вызов метода setDigit объекта cl аргументом shortDigit	21
21		вызов функции printArray аргументом значения поля symbol_pack объекта un	22
22		вызов функции printArray аргументом значения поля symbol_pack объекта cl	∅

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-5.

Рисунок 1 – Блок-схема алгоритма

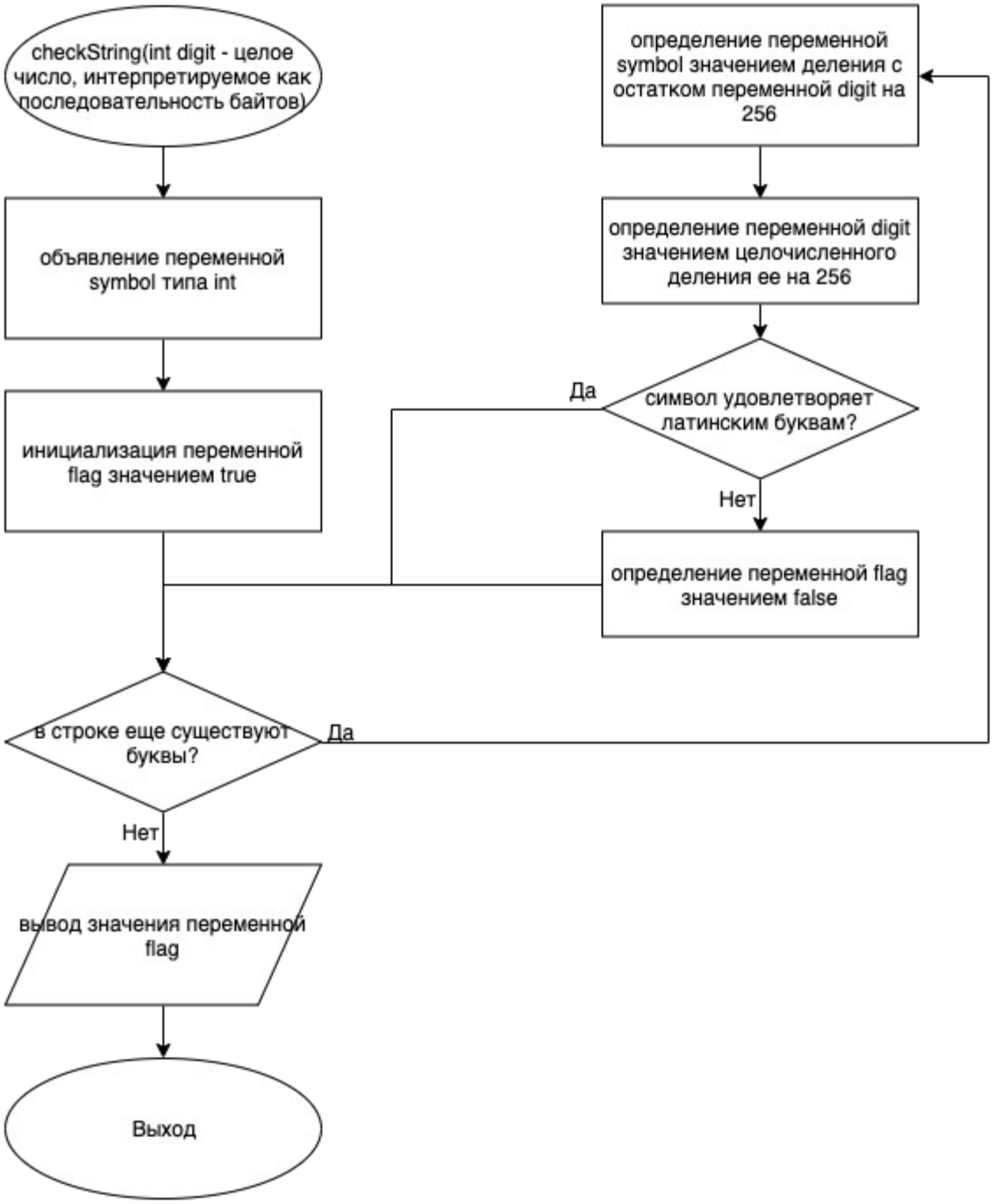

Рисунок 2 – Блок-схема алгоритма

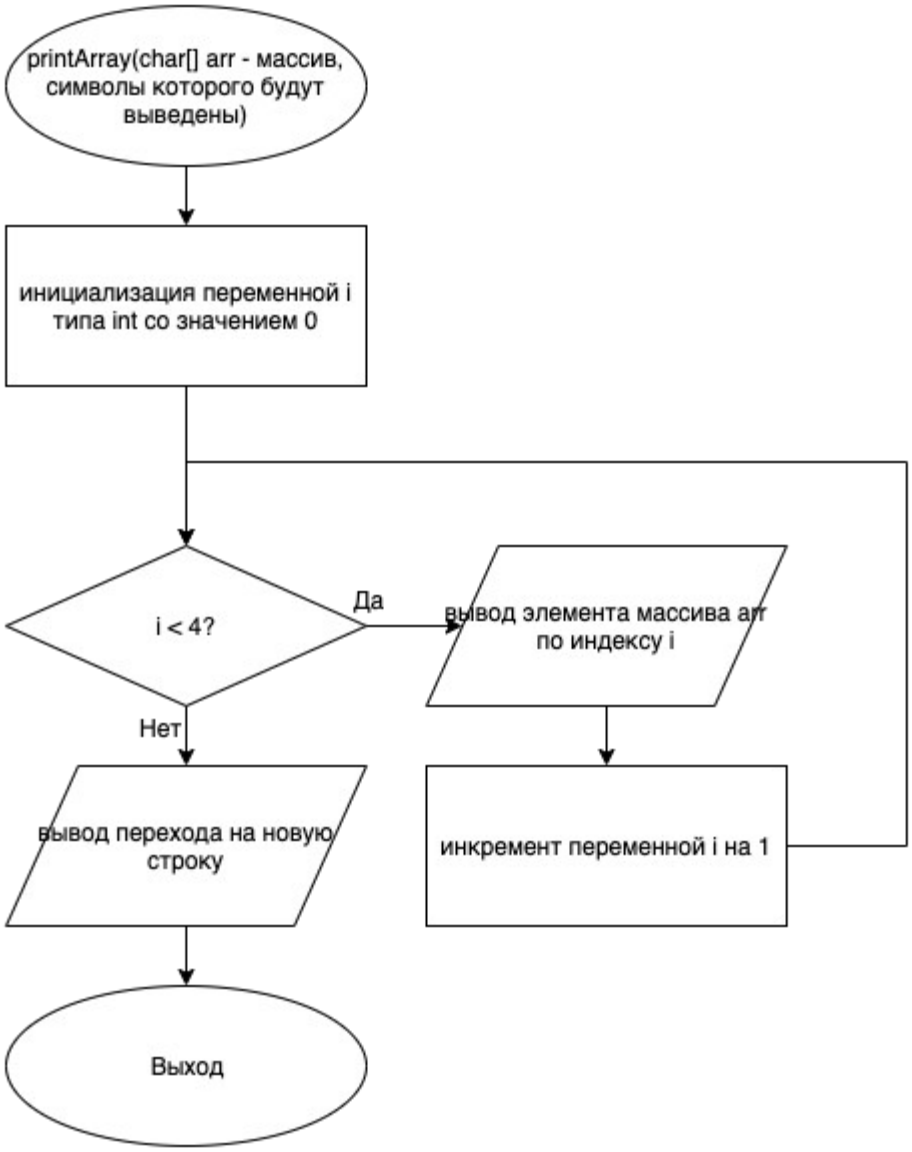

Рисунок 3 – Блок-схема алгоритма

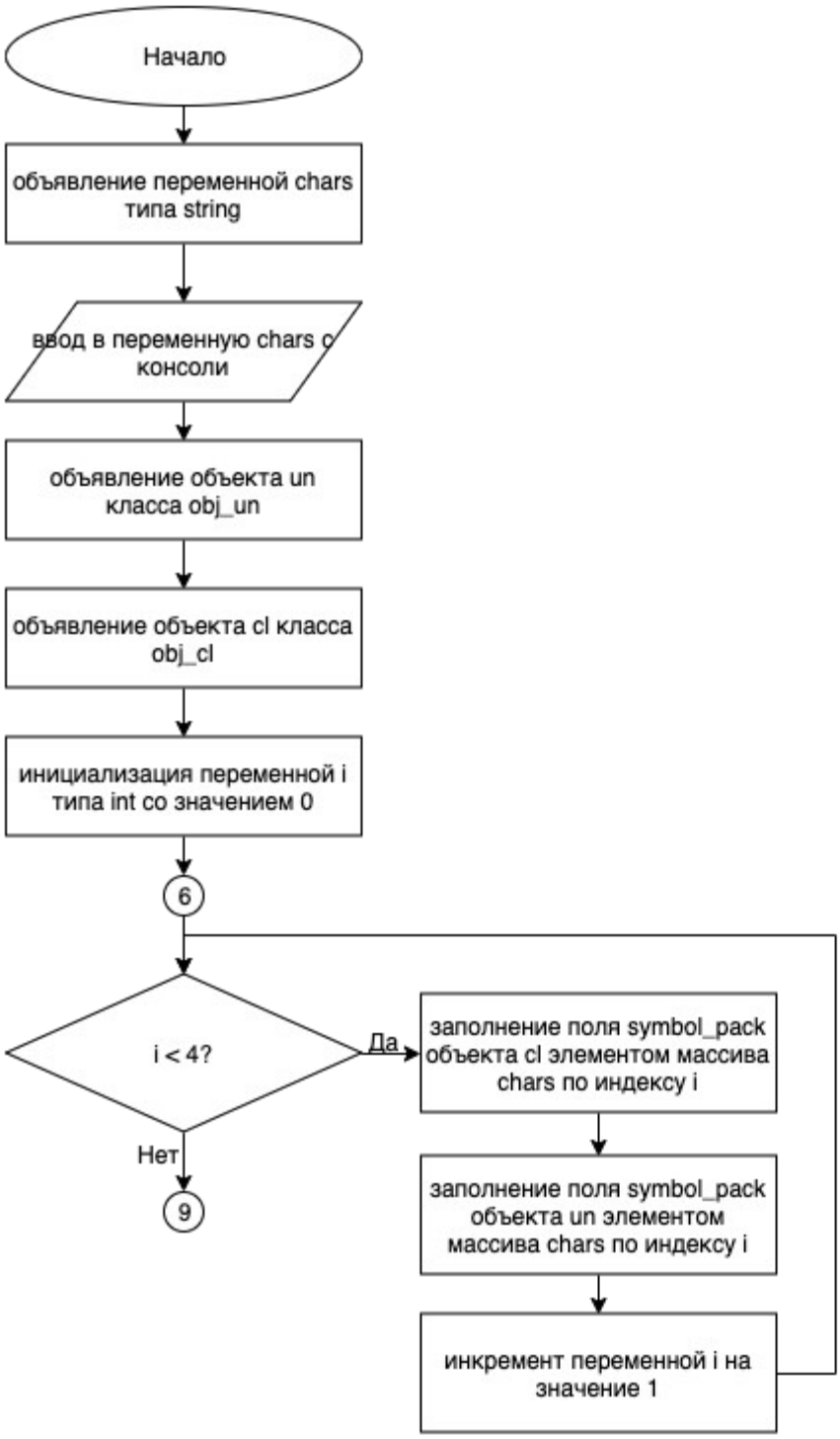

Рисунок 4 – Блок-схема алгоритма

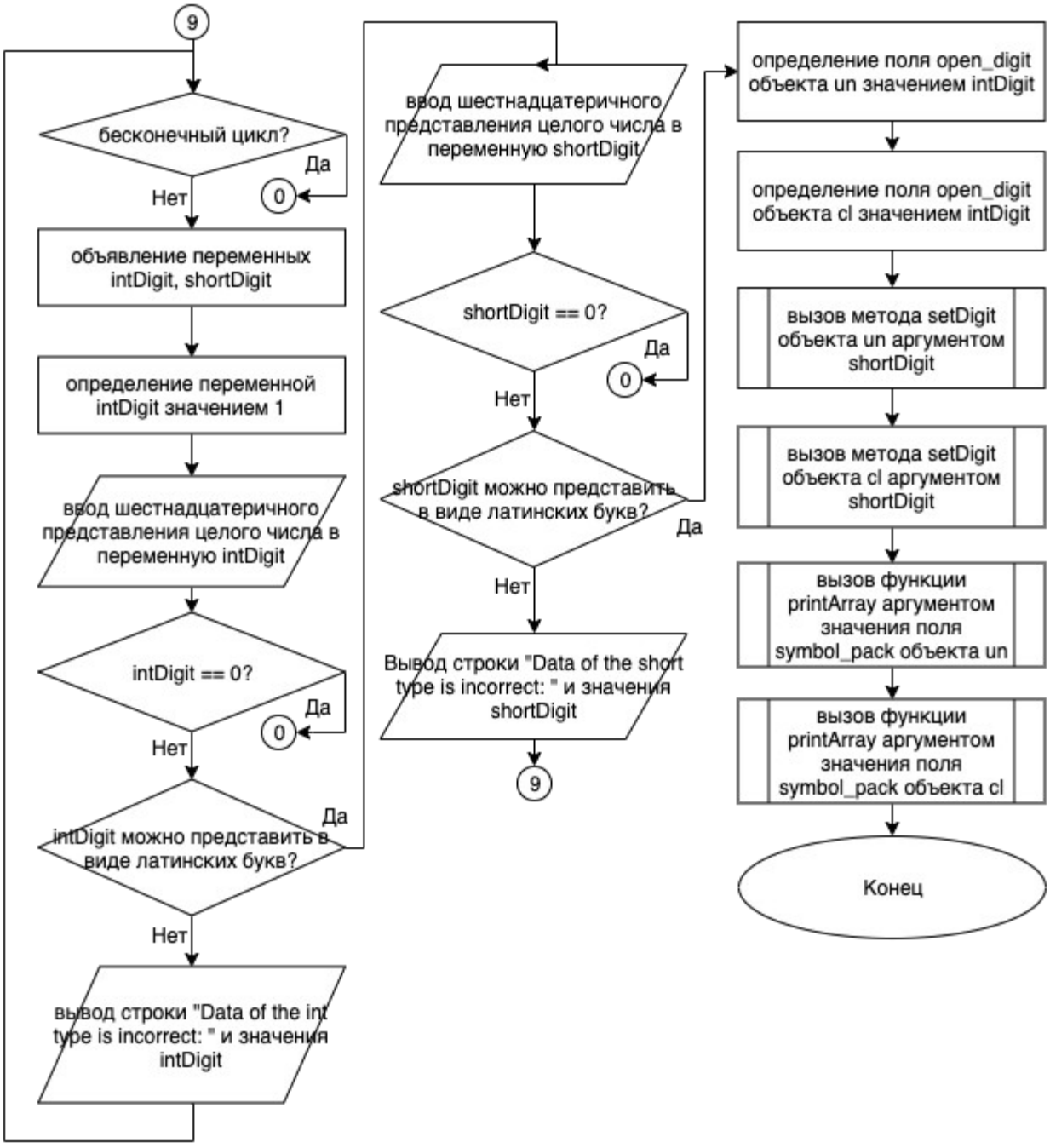

Рисунок 5 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

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

bool checkString(int digit){
   int symbol;
   bool flag = true;
   while (digit != 0){
      symbol = digit % 256;
      digit /= 256;
      if (!((65 <= symbol && symbol <= 90) || (97 <= symbol && symbol <=
122)))
         flag = false;
   }
   return flag;
}

void printArray(char arr[]) {
   for (int i = 0; i < 4; i++){
      cout << arr[i];
   }
   cout << endl;
}

int main()
{
   string chars;
   cin >> chars;

   obj_un un;
   obj_cl cl;

   for (int i = 0; i < 4; ++i){
      cl.symbol_pack[i] = chars[i];
      un.symbol_pack[i] = chars[i];
   }

   while (true) {
```

```
            int intDigit, shortDigit;
            intDigit = 0;

            cin >> hex >> intDigit;
            if (intDigit == 0) break;
            if (!checkString(intDigit)) {
               cout << hex << "Data of the int type is incorrect: " << intDigit <<
      endl;
               continue;
            }

            cin >> hex >> shortDigit;
            if (shortDigit == 0) break;
            if (!checkString(shortDigit)) {
               cout << hex << "Data of the short type is incorrect: " << shortDigit
      << endl;
               continue;
            }

            un.open_digit = intDigit;
            cl.open_digit = intDigit;
            un.setDigit(shortDigit);
            cl.setDigit(shortDigit);

            printArray(un.symbol_pack);
            printArray(cl.symbol_pack);
         }
         return(0);
      }
```

5.2 Файл obj_cl.cpp

Листинг 2 – obj_cl.cpp

```
      #include "obj_cl.h"

      void obj_cl::setDigit(short n){
         this->hidden_digit = n;
      }

      short obj_cl::getDigit(){
         return this->hidden_digit;
      }
```

5.3 Файл obj_cl.h

Листинг 3 – obj_cl.h

```
#ifndef __OBJ_CL__H
#define __OBJ_CL__H

class obj_cl{
   short hidden_digit;
public:
   int open_digit;
   char symbol_pack[4];

   void setDigit(short);
   short getDigit();
};

#endif
```

5.4 Файл obj_un.cpp

Листинг 4 – obj_un.cpp

```
#include "obj_un.h"

void obj_un::setDigit(short n){
   this->hidden_digit = n;
}

short obj_un::getDigit(){
   return this->hidden_digit;
}
```

5.5 Файл obj_un.h

Листинг 5 – obj_un.h

```
#ifndef __OBJ_UN__H
#define __OBJ_UN__H

union obj_un{
   int open_digit;
   char symbol_pack[4];
```

```
   void setDigit(short);
   short getDigit();
private:
   short hidden_digit;
};

#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 9.

Таблица 9 – Результат тестирования программы

Входные данные	**Ожидаемые выходные данные**	**Фактические выходные данные**
ABCD 0x41424344 0x4C4D 0x41424344 0x3C4D 0x41324344 0x4C4D 0x00	MLBA ABCD Data of the short type is incorrect: 3c4d Data of the int type is incorrect: 41324344	MLBA ABCD Data of the short type is incorrect: 3c4d Data of the int type is incorrect: 41324344

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).