

Здесь будет титульник, листай ниже

# СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	7
1.2 Описание выходных данных.....	7
2 МЕТОД РЕШЕНИЯ.....	9
3 ОПИСАНИЕ АЛГОРИТМОВ.....	10
3.1 Алгоритм конструктора класса Bus.....	10
3.2 Алгоритм метода getName класса Bus.....	10
3.3 Алгоритм функции main.....	10
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	13
5 КОД ПРОГРАММЫ.....	19
5.1 Файл Bus.cpp.....	19
5.2 Файл Bus.h.....	19
5.3 Файл main.cpp.....	20
6 ТЕСТИРОВАНИЕ.....	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	23

# 1 ПОСТАНОВКА ЗАДАЧИ

Разработать систему, которая моделирует движение автобусов по круговому маршруту с односторонним движением. Время движения между остановками одинаковая (одинаковый временной интервал). Остановки пронумерованы от 1 до n.

Сопоставить автобусу объект, у которого одно свойство строкового типа в закрытом доступе, для хранения номера автобуса.

Объект имеет конструктор с одним параметром строкового типа. Параметр содержит номер автобуса и его значение присваивается свойству с закрытым доступом.

Объект имеет метод в открытом доступе, который возвращает значение номера автобуса.

Расположение автобусов на маршруте моделировать (отобразить) ассоциативным контейнером, в котором значение ключа соответствует номеру остановки, которому ставится в соответствии указатель на объект автобуса, который находится на остановке. Допускаем, что на остановке может находиться только один автобус. Предполагается, количество автобусов меньше количества остановок.

Алгоритм конструирования и отработки системы:

1. Объявляется целочисленная переменная, для хранения количества остановок.
2. Объявляется целочисленная переменная, для хранения номера остановки.
3. Объявляется ассоциативный контейнер.
4. Объявляется целочисленная переменная, для хранения количества автобусов.

5. Объявляется строковая переменная, для хранения номера автобуса.
6. Вводится значение количества остановок.
7. В ассоциативном контейнере формируются элементы, которые соответствуют остановкам.
8. Вводится значение количества автобусов.
9. Цикл от единицы до количества автобусов.
  - 9.1. Вводится значение номере автобуса и значение номера остановки исходного расположения автобуса.
  - 9.2. Создание объекта автобус и размещение значение указателя на этот объект в контейнере согласно номеру остановки.
10. Конец цикла.
11. Начало цикла.
  - 11.1. Вводится целочисленное значение, которое равно количеству интервалов.
  - 11.2. Если значение интервала равно нулю, то выход из цикла.
  - 11.3. Реализуется перемещение автобусов по маршруту.
12. Коней цикла
13. Выводится в информация итогового расположения автобусов на маршруте построчно. Строка содержит: номер остановки и номер автобуса.
14. Очищается контейнер и удаляются объекты автобусов.
15. Завершается работа системы.

При сдаче предложите более оптимальное решение задачи. Обоснуйте решение.

## 1.1 Описание входных данных

Первая строка:

«целое число, количество остановок»

Вторая строка:

«целое число, количество автобусов»

Начиная с третьей строки, построчно, согласно количеству автобусов:

«строка, номер автобуса» «целое число, номер исходного расположения автобуса»

Начиная со следующей строки

«целое число, количество интервалов»

Последняя строка:

0

Пример ввода

```
10
3
77AP345 3
77AP115 9
77AP678 5
1
2
0
```

## 1.2 Описание выходных данных

Первая строка

stop bus

Начиная со второй строки, построчно

«номер остановки»

«номер автобуса»

### Пример вывода

stop	bus
2	77AP115
6	77AP345
8	77AP678

## 2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

- ассоциативный контейнер map;
- библиотека iostream;
- арифметические операторы;
- цикл for;
- цикл с предусловием while;
- операторы new и delete;
- условная конструкция if..else.

Класс Bus:

- свойства/поля:
  - поле номер автобуса:
    - наименование — number;
    - тип — string;
    - модификатор доступа — private;
- функционал:
  - метод Bus — конструктор;
  - метод getName — получение номера автобуса.

## 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 Bus

Функционал: конструктор.

Параметры: параметр name типа string.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Bus

№	Предикат	Действия	№ перехода
1		Присвоение полю number значения name	Ø

### 3.2 Алгоритм метода getName класса Bus

Функционал: получение номера автобуса.

Параметры: нет.

Возвращаемое значение: значение типа string.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода getName класса Bus

№	Предикат	Действия	№ перехода
1		Возвращение значения поля number	Ø

### 3.3 Алгоритм функции main

Функционал: главная функция программы.



Параметры: нет.

Возвращаемое значение: целое число, идентификатор работоспособности программы.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции *main*

№	Предикат	Действия	№ перехода
1		Объявление целочисленных переменных busStopsN и busStop	2
2		Объявление переменной busStops типа map<int, Bus*>	3
3		Объявление целочисленной переменной busesN	4
4		Объявление переменной busName типа string	5
5		Ввод значения busStopsN	6
6		Инициализация целочисленной переменной i со значением 1	7
7	i <= busStopsN?	Присвоение элементу busStops по ключу i значения nullptr	8
			9
8		Инкремент i	7
9		Ввод значения busesN	10
10		Инициализация целочисленной переменной i со значением 1	11
11	i <= busStopsN	Ввод значений busName и busStop	12
			14
12		Создание объекта класса Bus с передачей в конструктор значения busName и присваивание указателя на этот	13
13		Инкремент i	11
14		Объявление переменной busStops2 типа	15

№	Предикат	Действия	№ перехода
		map<int, Bus*>	
15		Объявление целочисленной переменной interval	16
16		Ввод значения interval	17
17	interval == 0?		24
		Инициализация целочисленной переменной i со значением 1	18
18	i <= busStopsN		19
			23
19	busStops[i] != nullptr?	Инициализация целочисленной переменной newBusStop значением (i+interval)%busStopsN	20
			22
20	newBusStop == 0?	Присвоение newBusStop значения busStopsN	21
			21
21		Присваивание busStops2[newBusStop] значения busStops[i]	22
22		Инкремент i	18
23		Присваивание busStops значения busStops2	14
24		Вывод строки "stop bus"	25
25		Инициализация целочисленной переменной i значением 1	26
26	i <= busStopsN		27
			∅
27	busStops[i] != nullptr?	Вывод i и busStops[i]->getName()	28
			28
28		Инкремент i	26

## 4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-6.

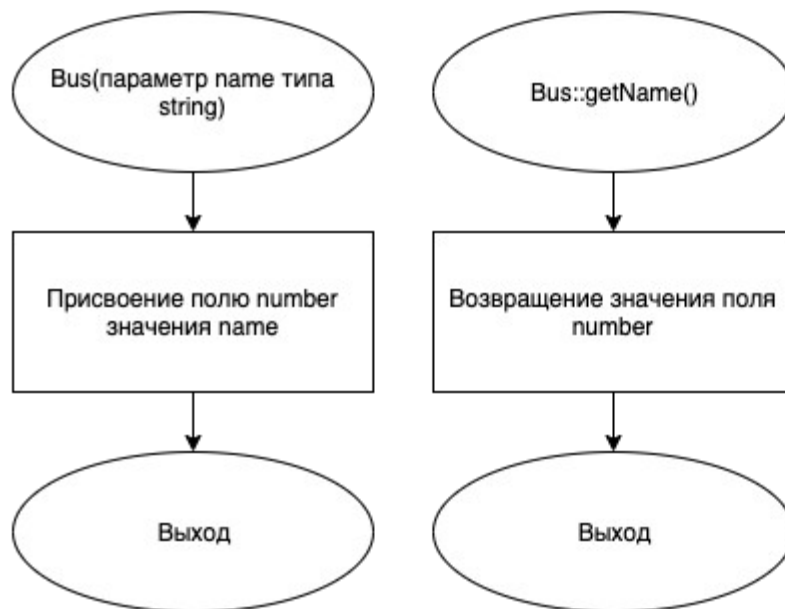


Рисунок 1 – Блок-схема алгоритма

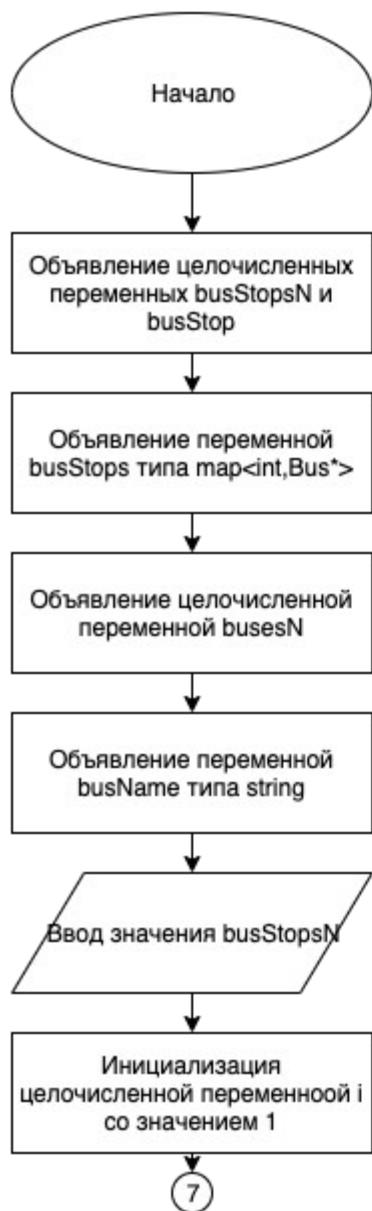


Рисунок 2 – Блок-схема алгоритма

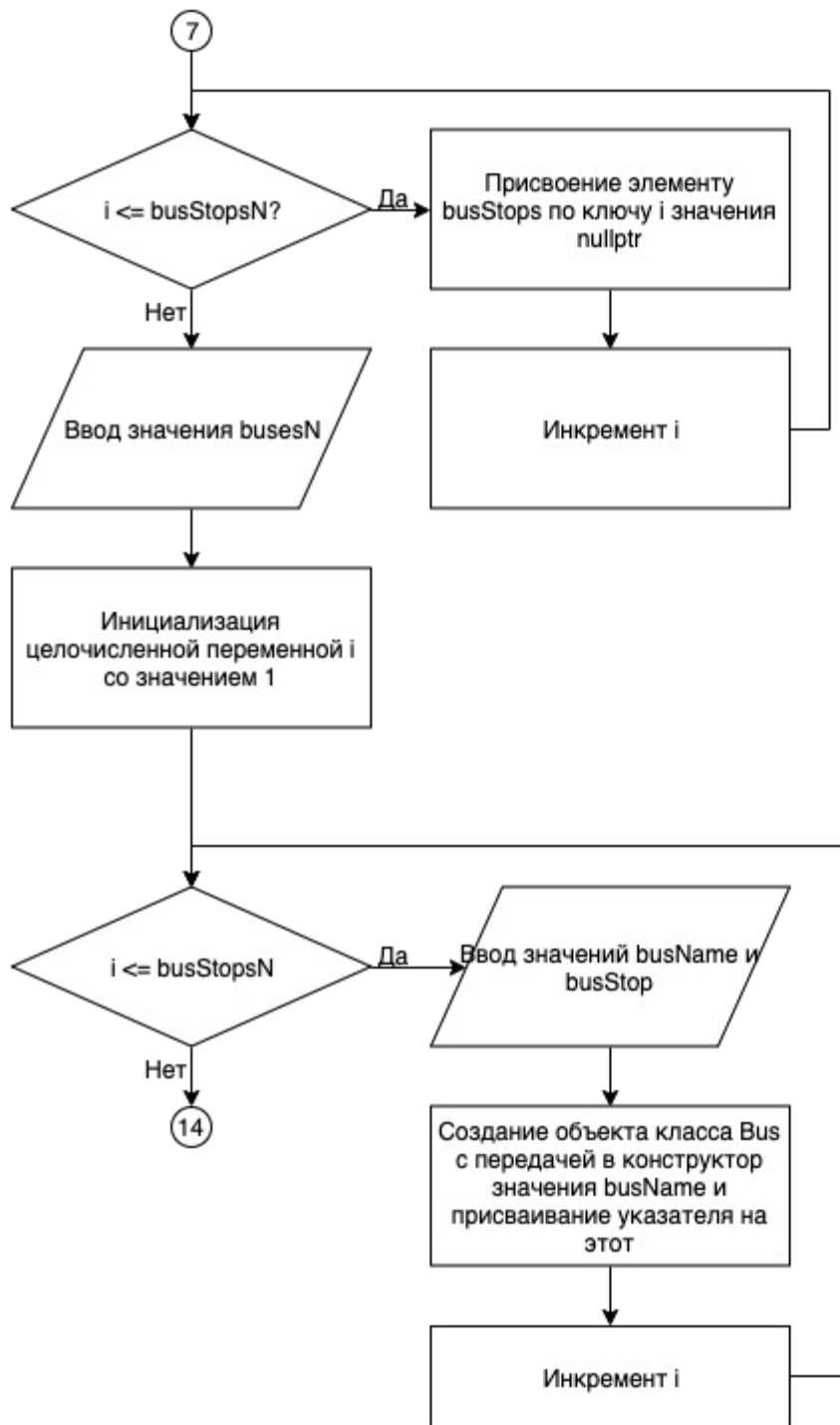


Рисунок 3 – Блок-схема алгоритма

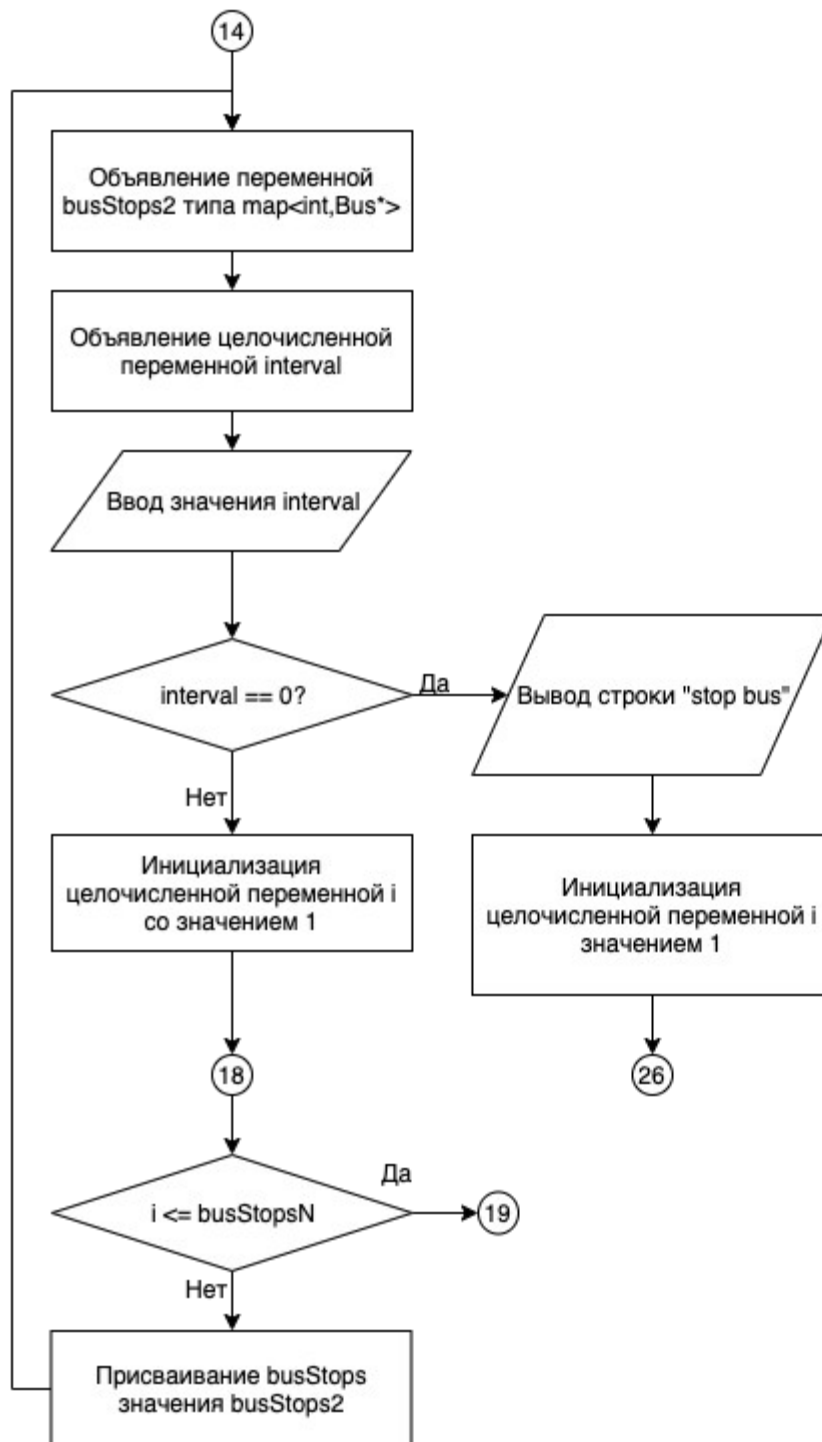
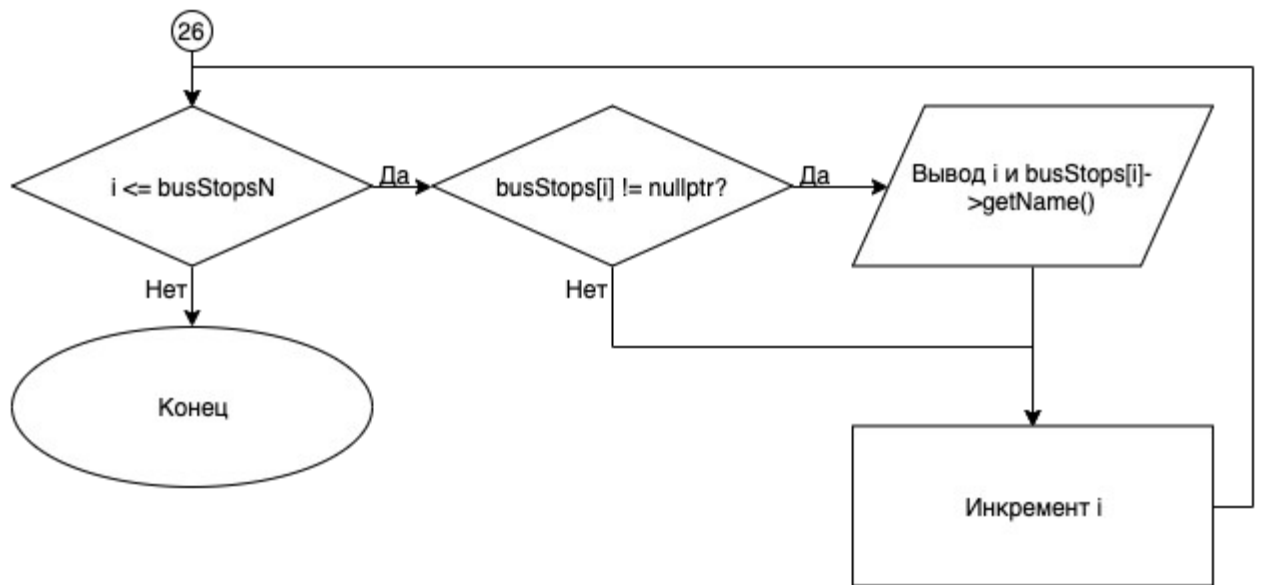


Рисунок 4 – Блок-схема алгоритма



**Рисунок 5 – Блок-схема алгоритма**

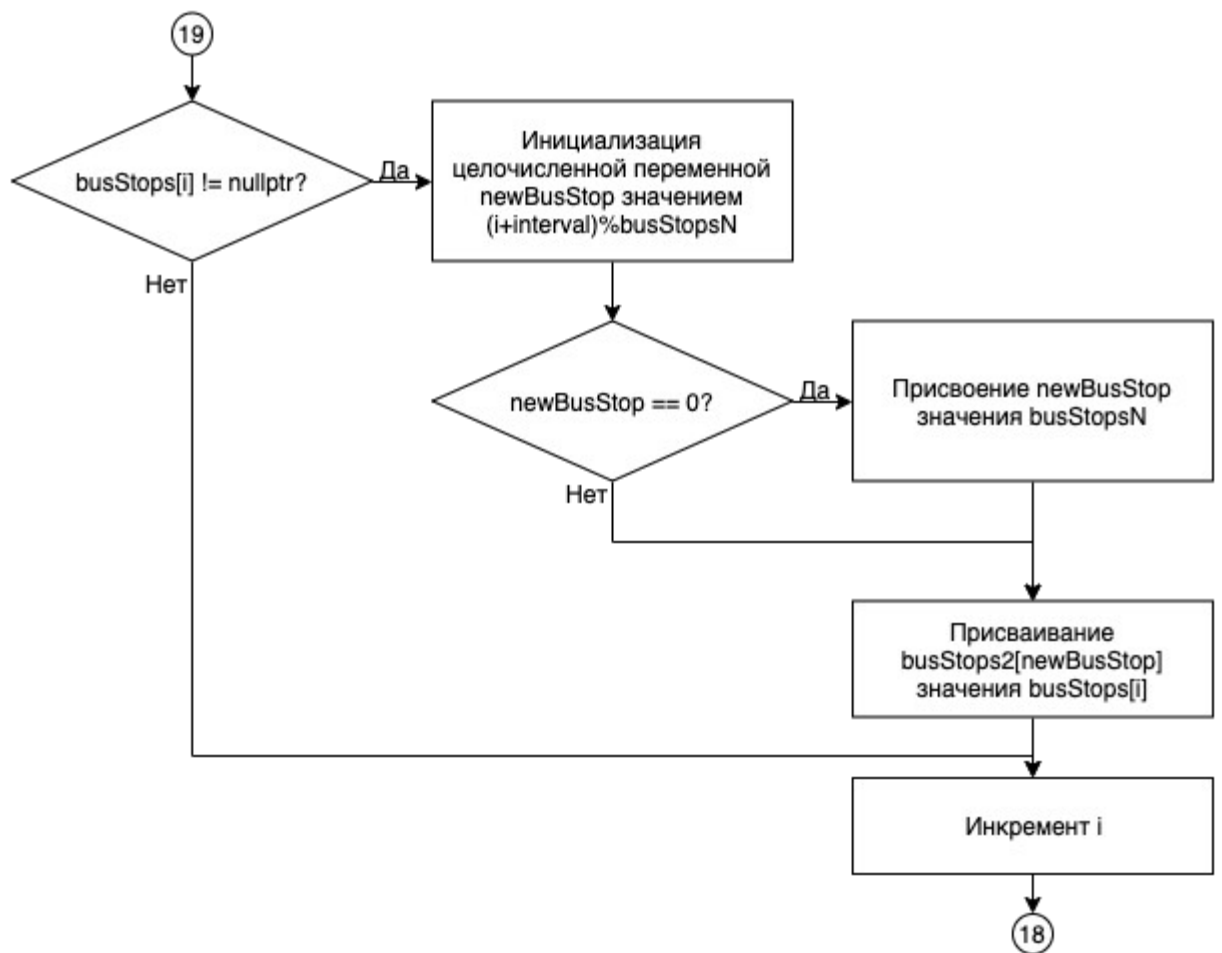


Рисунок 6 – Блок-схема алгоритма



## 5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

### 5.1 Файл Bus.cpp

*Листинг 1 – Bus.cpp*

```
#include "Bus.h"
#include <string>
using namespace std;

Bus::Bus(string name){
    this->number = name;
}

string Bus::getName(){
    return number;
}
```

### 5.2 Файл Bus.h

*Листинг 2 – Bus.h*

```
#ifndef __BUS__H
#define __BUS__H
#include <string>
using namespace std;

class Bus{
    string number;
public:
    string getName();
    Bus(string name);
};

#endif
```

## 5.3 Файл main.cpp

Листинг 3 – main.cpp

```
#include <iostream>
#include <string>
#include <map>
#include "Bus.h"
using namespace std;

int main()
{
    int busStopsN;
    int busStop;
    map<int, Bus*> busStops;
    int busesN;
    string busName;

    cin >> busStopsN;
    for (int i = 1; i<=busesN; i++){
        busStops[i] = nullptr;
    }

    cin >> busesN;
    for (int i = 1; i<=busesN; i++){
        cin >> busName >> busStop;
        busStops[busStop] = new Bus(busName);
    }

    while (true) {
        map<int, Bus*> busStops2;
        int interval;
        cin >> interval;
        if (interval == 0){
            break;
        }
        for (int i=1; i<= busStopsN; i++){
            if (busStops[i] != nullptr){
                int newBusStop = (i+interval)%busStopsN;
                if (newBusStop == 0){
                    newBusStop = busStopsN;
                }
                busStops2[newBusStop] = busStops[i];
            }
        }
        busStops = busStops2;
    }

    cout << "stop    bus" << endl;

    for (int i = 1; i<=busStopsN; i++){
        if (busStops[i] != nullptr){
            cout << i << "    ";
            cout << busStops[i]->getName() << endl;
        }
    }
}
```

```
    }  
  }  
  return(0);  
}
```

## 6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 4.

Таблица 4 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
10 3 77AP345 3 77AP115 9 77AP678 5 1 2 0	stop bus 2 77AP115 6 77AP345 8 77AP678	stop bus 2 77AP115 6 77AP345 8 77AP678

## 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/methodichescoc\\_posobie\\_dlya\\_laboratornyh\\_rabot\\_3.pdf](https://mirea.aco-avvora.ru/student/files/methodichescoc_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](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).