Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

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 присваиваться 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 ставиться в соответствии указатель на объект автобуса, который находится на остановке. Допускаем, что на остановке может находится 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. Вводиться значение количества остановок.

7. В ассоциативном контейнере формируются элементы, которые соответствуют остановкам.

8. Вводиться значение количества автобусов.

9. Цикл от единицы до количества автобусов.

9.1. Вводиться 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. Вводиться 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:

- функция main для определения входной точки программы;
- заголовочный файл;
- класс;
- указатель;
- словарь;
- библиотека ввода-вывода.

Класс Bus:

- свойства/поля:
    - поле хранит номер автобуса:
        - наименование — m_number;
        - тип — std::string;
        - модификатор доступа — private;
- функционал:
    - метод Bus — инициализирует приватное поле m_number;
    - метод GetNumber — возвращает значение 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: Инициализирует приватное поле m_number.

Параметры: std::string number.

Алгоритм конструктора представлен в таблице 1.

Таблица 1 – Алгоритм конструктора класса Bus

№	Предикат	Действия	№ перехода
1		Инициализация m_number = number	∅

3.2 Алгоритм метода GetNumber класса Bus

Функционал: Возвращает значение номера автобуса.

Параметры: нет.

Возвращаемое значение: std::string.

Алгоритм метода представлен в таблице 2.

Таблица 2 – Алгоритм метода GetNumber класса Bus

№	Предикат	Действия	№ перехода
1		Возврат m_number	∅

3.3 Алгоритм функции main

Функционал: Входная точка программы.

Параметры: нет.

Возвращаемое значение: int.

Алгоритм функции представлен в таблице 3.

Таблица 3 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Объявление целочисленных переменных stopsCount, stopNumber, busesCount, intervals	2
2		Объявление словаря route	3
3		Объявление переменной строкового типа busNumber	4
4		Ввод stopsCount	5
5		Ввод busesCount	6
6		Инициализация int i = 0	7
7	i < busesCount	Ввод busNumber, stopNumber	8
			10
8		Инициализация объекта bus класса Bus с помощью оператора new с передачей busNumber в качестве параметра	9
9		Присвоение route[stopNumber] = bus	7
10	true	Ввод intervals	11
			19
11	intervals = 0		19
			12
12		Объявление std::map<int, Bus*> newRoute	13
13	Есть следующий элемент в route	Перебор for (auto pair : route)	15
			18
14		Получение currentStop = pair.first, bus = pair.second	15
15		Вычисление newStop = (currentStop + intervals) %	16

№	Предикат	Действия	№ перехода
		stopsCount	
16	newStop = 0	Присвоение newStop = stopsCount	17
			17
17		Присвоение newRoute[newStop] = bus	13
18		Присвоение route = newRoute	10
19		Вывод "stop bus"	20
20	Есть следующий элемент в route	Перебор for (auto pair : route)	21
			∅
21		Вывод pair.first и pair.second->GetNumber()	22
22		Освобождение памяти: delete pair.second	20

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-4.

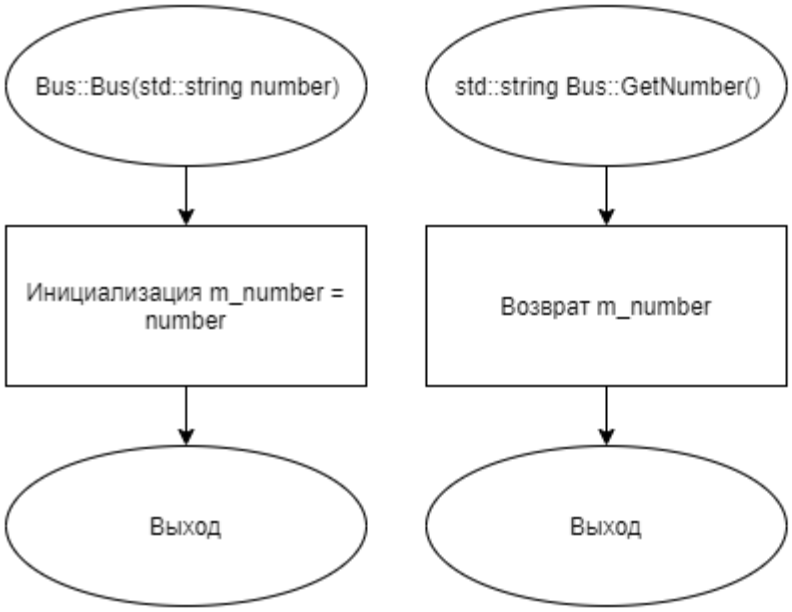

Рисунок 1 – Блок-схема алгоритма

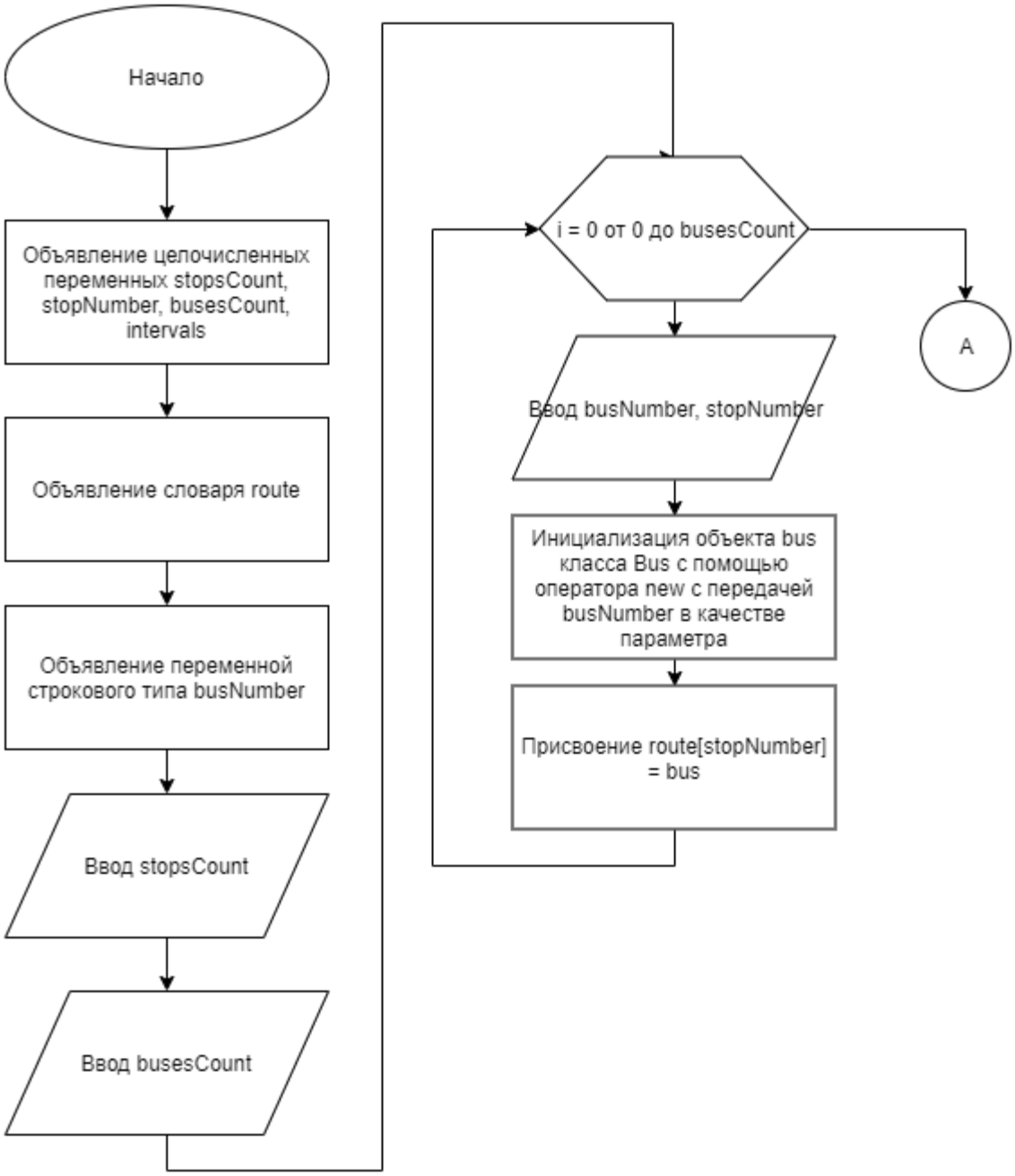

Рисунок 2 – Блок-схема алгоритма

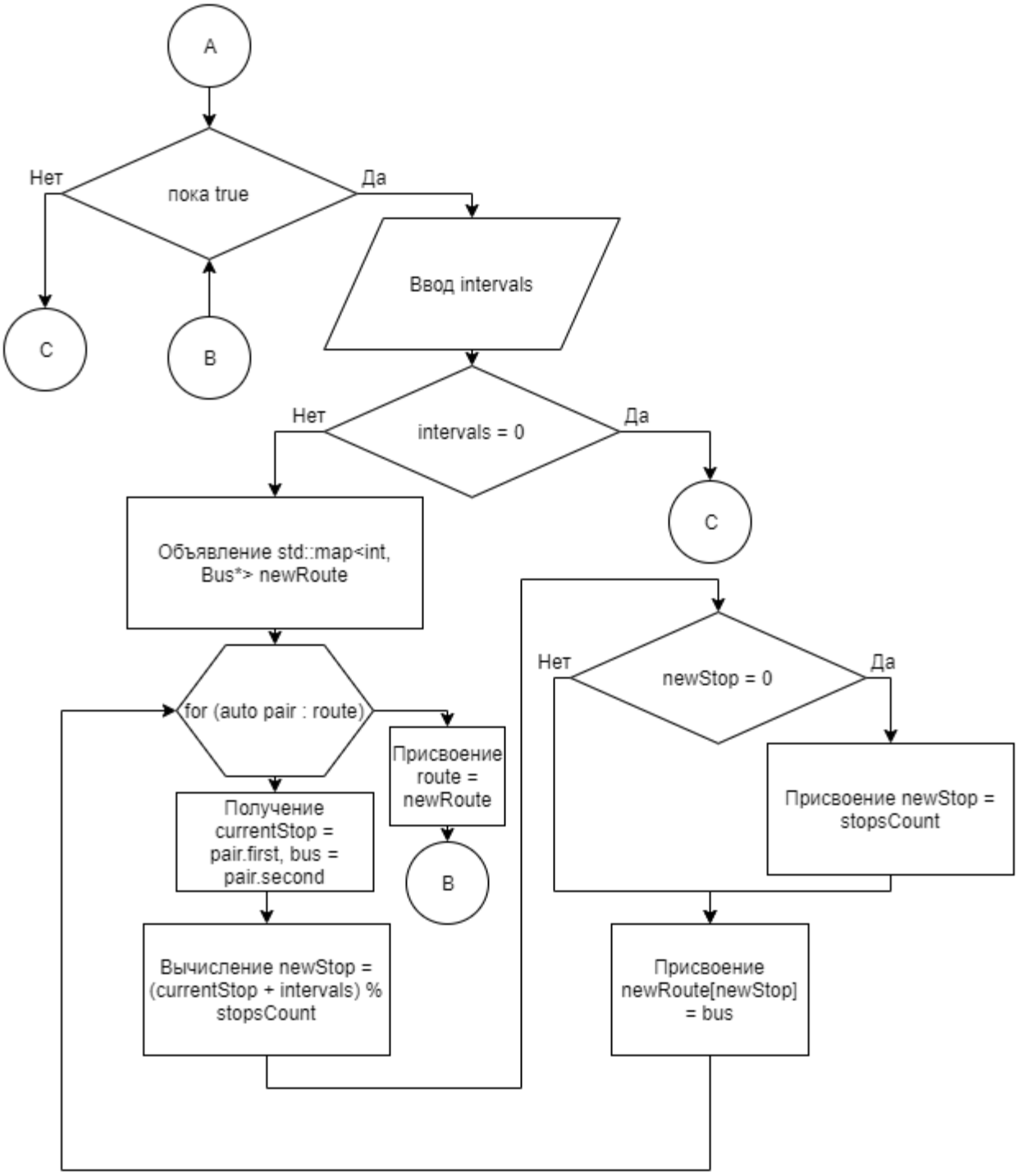

Рисунок 3 – Блок-схема алгоритма

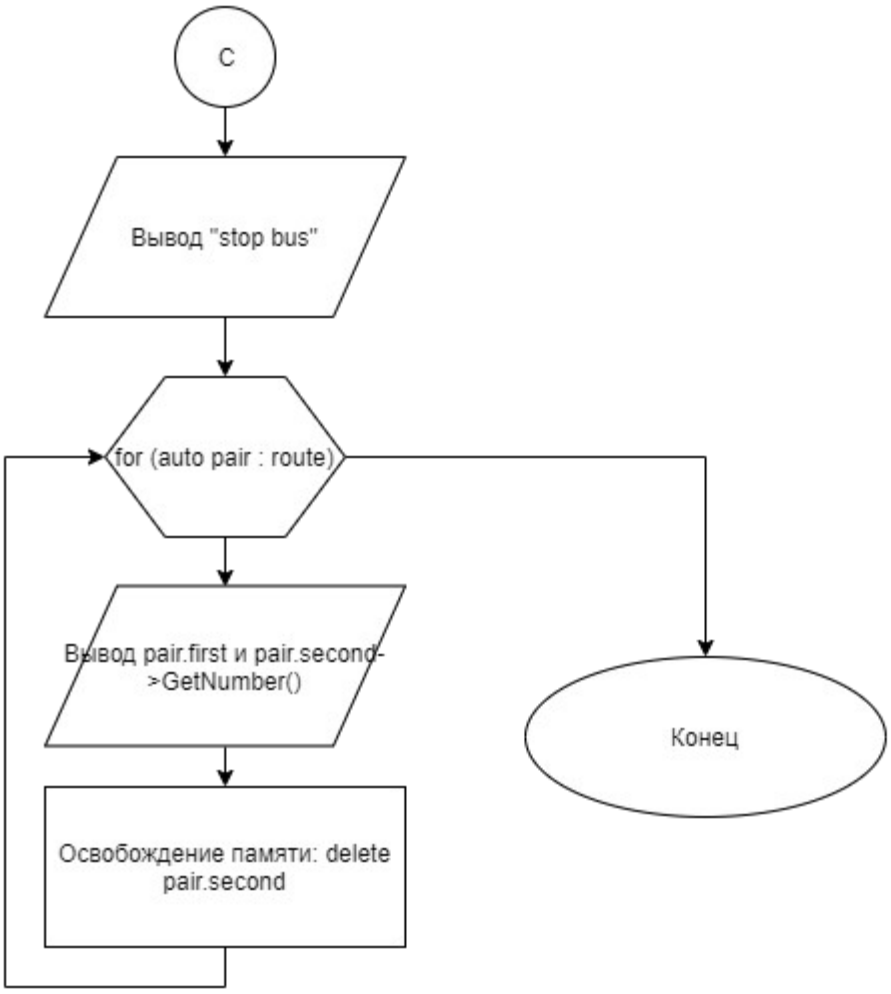

Рисунок 4 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл Bus.cpp

Листинг 1 – Bus.cpp

```
#include "Bus.h"

Bus::Bus(std::string number)
{
   m_number = number;
}

std::string Bus::GetNumber()
{
   return m_number;
}
```

5.2 Файл Bus.h

Листинг 2 – Bus.h

```
#ifndef __BUS__H
#define __BUS__H

#include <string>

class Bus
{
   std::string m_number;
public:
   Bus(std::string number);
   std::string GetNumber();
};

#endif
```

5.3 Файл main.cpp

Листинг 3 – main.cpp

```
#include "Bus.h"
#include <iostream>
#include <map>

int main()
{
   int stopsCount, stopNumber, busesCount, intervals;
   std::map<int, Bus*> route;
   std::string busNumber;

   std::cin >> stopsCount;
   std::cin >> busesCount;

   for (int i = 0; i < busesCount; i++)
   {
      std::cin >> busNumber >> stopNumber;
      Bus* bus = new Bus(busNumber);
      route[stopNumber] = bus;
   }

   while (true)
   {
      std::cin >> intervals;
      if (intervals == 0) break;

      std::map<int, Bus*> newRoute;

      for (auto pair : route)
      {
         int currentStop = pair.first;
         Bus* bus = pair.second;
         int newStop = (currentStop + intervals) % stopsCount;
         if (newStop == 0) newStop = stopsCount;

         newRoute[newStop] = bus;
      }

      route = newRoute;
   }

   std::cout << "stop    bus" << std::endl;
   for (auto pair : route)
   {
      std::cout << pair.first << "         " << pair.second->GetNumber() <<
std::endl;
      delete pair.second;
   }

   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-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).