|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Робототехника и комплексная автоматизация (РК)

КАФЕДРА Системы автоматизированного проектирования (РК-6)

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**по курсу объектно-ориентированное программирование**

Студент Камалов Антон Павлович

Группа РК6-25Б

Тип задания Лабораторная работа №4

Вариант А08

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Камалов А.П.**

*подпись, дата фамилия, и.о.*

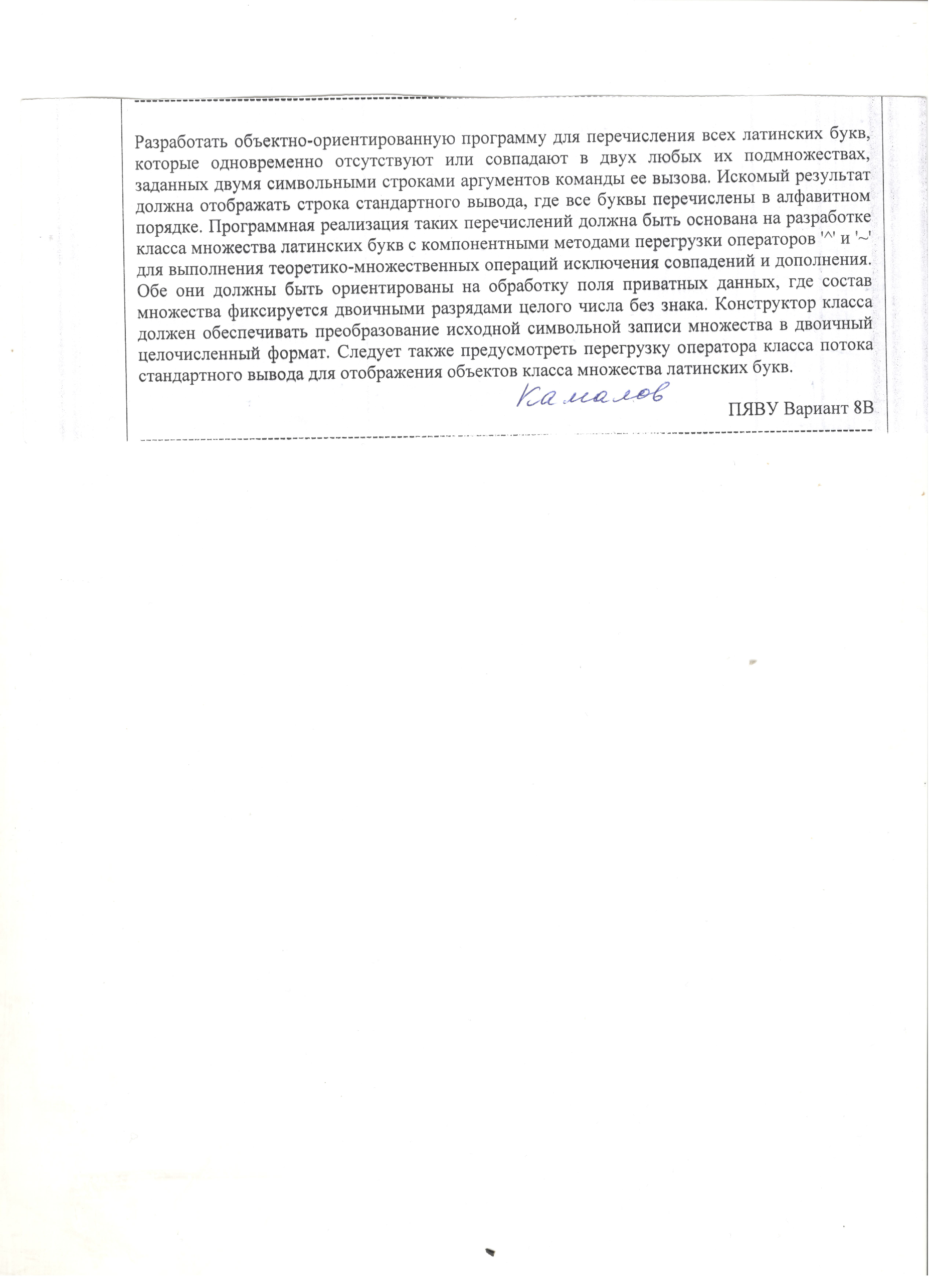
Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кузьмина И.А.**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2023 г.*

**Задание**

****

**Описание программы**

**Входные данные:** символьная строка, состоящая из букв латинского алфавита.

**Выходные данные:** количество согласных латинских букв, отсутствующих в строке.

**Класс Alpha**

*Информационные поля:*

set (unsigned) – битовая маска множества букв;

*Конструкторы и деструкторы:*

Alpha (const char \*, const char \*) – с параметрами;

Alpha () – по умолчанию;

*Методы:*

Alpha operator & (const Alpha&) – публичный метод перегрузки оператора побитового И (&);

operator char\*() – публичный метод перегрузки оператора приведения тип (char\*);

int pop(unsigned) – публичный метод быстрого подсчёта единичных разрядов в бинарном коде;

int operator () (Alpha&, Alpha&) – публичный метод перегрузки оператора вызова функции ().

**Общий алгоритм решения:**

1. Проверяем количество аргументов командной строки. Если их меньше 2, то программа завершает работу с кодом возврата -1;
2. Создаём 3 объекта класса Alpha – str (битовая маска, содержащая буквы входной строки), sogl (битовая маска, содержащая все согласные буквы), H (пустая битовая маска);
3. Используя побитовое И для класса Alpha (с помощью перегруженного оператора «&»), получаем новый объект класса Alpha set\_sogl = str ^ sogl;
4. Вычисляем расстояние Хэмминга между set\_sogl и sogl (с помощью перегруженного оператора «( )»)
5. Преобразовываем str и sogl в строки, выводим их в буфер стандартного вывода с помощью перегруженного оператора (char\*). Также выведем на экран полученное расстояние Хэмминга.

**Список литературы**

1. Пол Айра. Объектно-ориентированное программирование на C++: пер. с англ. 2-е изд. СПБ.: Невский Диалект; М.: Издательство БИНОМ, 2001. 462 с. [Pohl Ira. Object-Oriented Programming Using C++. 2nd ed. Addison-Wesley, 1996. 576 p.].
2. Страуструп Б. Язык программирования C++: пер. с англ / под ред. Н.Н. Мартынова. Специальное изд. М.: Бином, 2011. 1035 c. [Stroustrup B. The C++ Programming Language. Special ed. Addison-Wesley, 2000. 1029p.].
3. Шилдт Г. Самоучитель C++. 3-е изд. СПБ.: БХВ-Петербург, 2002. 688 с. [Schildt H. Teach Yourself C++. 3d ed. McGraw-Hill, 1998. 768 p.].
4. Волосатова Т.М., Родионов С.В. Объектно-ориентированное программирование на С++. Режим доступа: http://bigor.bmstu.ru/?cnt/?doc=VU/base.cou (дата обращения 20.03.2023).

**Приложение 1**

Текст программы

#include <iostream>

#include <cstring>

class Alpha{

unsigned \_bin;

public:

Alpha() : \_bin(0) {};

Alpha (const char \*);

Alpha operator &(Alpha&);

operator char\*();

int pop(unsigned b);

int operator () (Alpha&, Alpha&);

};

Alpha::Alpha(const char \*s){

\_bin = 0;

while (\*s){

\_bin|= (1 << (tolower(\*s) - 'a'));

s++;

}

}

// эффективный метод подсчёта единичных разрядов

int Alpha::pop(unsigned b){

int i = 0;

while (b != 0) {

b = b & (b - 1);

i++;

}

return i;

}

Alpha Alpha::operator &(Alpha &n){

Alpha result;

result.\_bin = \_bin & n.\_bin;

return result;

}

int Alpha::operator () (Alpha &m, Alpha &n){

return pop(m.\_bin ^ n.\_bin);

}

Alpha::operator char\*(){

static char s[32];

unsigned b = \_bin;

int i = 0, j = 0;

while (b > 0){

if (b & 1) {

s[j++] = 'a' + i;

}

i++;

b >>= 1;

}

s[j] = '\0';

return s;

}

int main(int argc, char \*\*argv){

using std::cout;

if (argc < 2) return -1;

Alpha str(argv[1]);

Alpha sogl("bcdfghjklmnpqrstvwxz");

Alpha H;

Alpha set\_sogl = str & sogl;

int d = H(set\_sogl, sogl);

cout << "<(" << (char\*)str << ", " << (char\*)sogl << ") = " << d << '\n';

}

**Приложение 2**

Результат работы программы

