|  |  |
| --- | --- |
|  | **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 №3

Вариант 10А

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

*подпись, дата фамилия, и.о.*

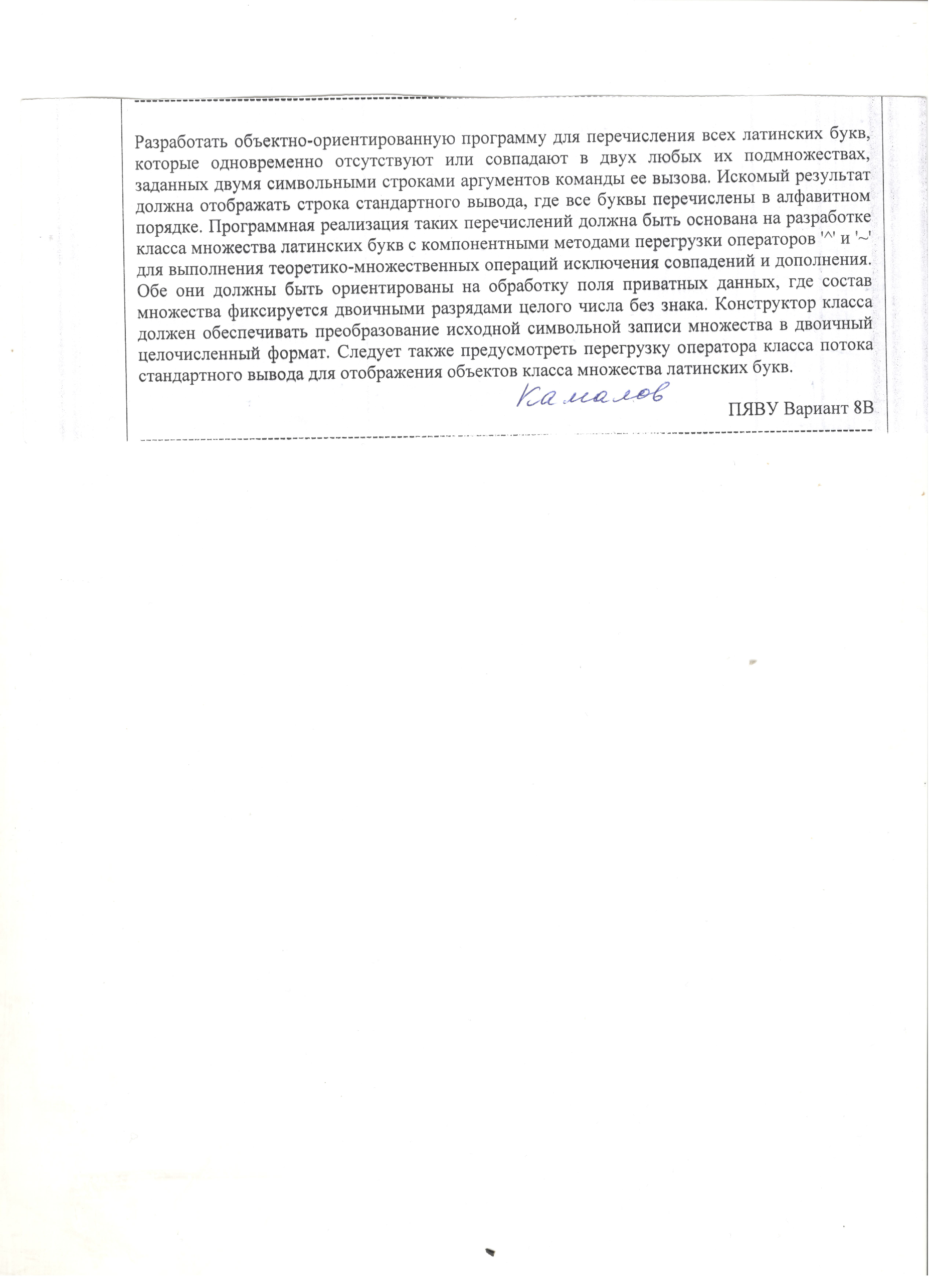
Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Козов А.В.**

*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) – битовую маска множества букв;

*Конструкторы и деструкторы:*

Alpha (char \*) – с параметром;

Point () – по умолчанию;

*Методы:*

Alpha operator ^ (const Alpha&) – публичный метод перегрузки оператора побитового исключающего ИЛИ (^);

Alpha operator ~ () – публичный метод перегрузки оператора побитового НЕ (~);

friend std::ostream& operator << (std::ostream&, const Alpha&)– дружественный метод вывода содержимого множества букв в стандартный поток вывода;

**Общий алгоритм решения:**

1. Проверяем количество аргументов командной строки. Если их меньше 3, то программа завершает работу с кодом возврата -1;
2. Создаём 2 объекта класса Alpha – s1 и s2 (представляют собой 32-битное число, где 1 означает наличие символа в строке;
3. Используя инвертирование и побитовое исключающее ИЛИ для класса Alpha (с помощью перегруженных операторов «~» и «^»), получаем новый объект класса Alpha result = ~ (s1 ^ s2);
4. Преобразовываем result в строку и выводим в буфер стандартного вывода с помощью перегруженного оператора «<<».

**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 11.03.2023).

**Приложение 1**

Текст программы

#include <iostream>

class Alpha {

private:

unsigned set;

public:

Alpha(){set = 0;}

Alpha(char \*s);

Alpha operator^(const Alpha&);

Alpha operator~();

friend std::ostream& operator<<(std::ostream&, const Alpha&);

};

Alpha::Alpha(char \*s){

set = 0;

while(\*s){

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

s++;

}

}

Alpha Alpha::operator~(){

Alpha result;

result.set = ~ set;

return result;

}

Alpha Alpha::operator^(const Alpha& other) {

Alpha result;

result.set = (set ^ other.set);

return result;

}

std::ostream& operator<<(std::ostream& os, const Alpha& s) {

unsigned bit = 1;

for (int i = 0; i < 26; i++) {

if ((s.set & bit) > 0) {

os << (char)('a' + i);

}

bit = bit << 1;

}

return os;

}

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

if (argc < 3) return -1;

Alpha s1(argv[1]);

Alpha s2(argv[2]);

Alpha result = ~ (s1 ^ s2);

std::cout << "Result: " << result << std::endl;

return 0;

}

**Приложение 2**

Результат работы программы

