

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения

вычислительной техники и автоматизированных

систем

Лабораторная работа №4

по дисциплине: ООП

тема: «Классы»

Выполнил: студент группы

ПВ-233

Мороз Роман Алексеевич

Проверили:

Белгород 2025

Лабораторная работа № 4

Классы

Цель работы: приобретение практических навыков создания класса на языке C++.

Задание: выполнить построение диаграммы объектов (не менее 7-ми объектов) в соответствии с заданием варианта (табл. 1). По заданному описанию класса в соответствии с вариантом создать класс на языке C++ и переопределить указанные операции.

Контрольные вопросы:

1. Класс это?
2. Способы создания объектов и виды конструкторов?
3. Модификаторы доступа и их использование.
4. UML для описания класса?
5. Отличие метода класса от функции.
6. Виды операторов в C++.
7. Friend функции.
8. Перегрузка операторов и ее виды.
9. Оператор приведения типов.

Пример выполнения второго задания:

Перегрузка оператора бинарного сложения для класса Car, которая при сложении двух экземпляров класса возвращает новый, имя в котором берется от первого параметра, а скорость равняется сумме скоростей экземпляров класса.

```
#include <iostream>
```

```
class Car {  
    float speed;  
    float mass;  
    char name[10];  
  
public:  
    Car(char* n){  
        for(int i=0; i<10; i++)  
            name[i]=n[i];  
        speed=10;  
    };  
};
```

```

void printCar(){
std::cout<<name<<' '<< speed <<std::endl;
};

friend const Car operator+ (const Car& a1, const Car& a2){

    char *name = new char[10];
    for(int i=0; i<10; i++)
        name[i]=a1.name[i];
    Car a3 (name);
    a3.speed=a1.speed+a2.speed;
    return a3;

}

};

int main ()
{   char *name = new char[10];
    name = "nissan";
    Car almera(name);
    almera.printCar();

    name = "nissan1";
    Car almera1(name);
    almera1.printCar();

    name = "nissan2";
    Car almera2(name);
    almera2.printCar();

    almera = almera1+almera2;
    almera.printCar();
    return 0;
}

```

Табл. 1

№	Задание 1	Задание 2
1	Выполнить построение объектной модели следующей	Создать класс Point, разработав следующие элементы класса: а. Поля:

	<p>предметной области: “средство построения растрового изображения”. При чем обязательно выделить отдельный объект “Точка”.</p>	<p>int x, y; int color;</p> <p>б. Конструкторы, позволяющие создать экземпляр класса: с нулевыми координатами; с заданными координатами.</p> <p>в. Методы, позволяющие: вывести координаты точки на экран; рассчитать расстояние от начала координат до точки; переместить точку на плоскости на вектор (a, b).</p> <p>г. Перегрузить (переопределить) следующие операторы: присвоение; сравнение(равенство);</p>
2	<p>Выполнить построение объектной модели следующей предметной области: “система триангуляции изображений”</p>	<p>Создать класс Triangle, разработав следующие элементы класса:</p> <p>а. Поля: int a, b, c;</p> <p>б. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.</p> <p>в. Методы, позволяющие: вывести длины сторон треугольника на экран; рассчитать периметр треугольника; рассчитать площадь треугольника.</p> <p>г. Перегрузить (переопределить): сравнение(больше); сложение(унарный, префиксное).</p>
3	<p>Выполнить построение объектной модели следующей предметной области: “Простой тетрис”. (простым тетрисом называется такой тетрис, в котором нет ломаных фигур, т.е. все фигуры квадраты,</p>	<p>Создать класс Rectangle, разработав следующие элементы класса:</p> <p>а. Поля: int a, b;</p> <p>б. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.</p> <p>в. Методы, позволяющие: вывести длины сторон прямоугольника на экран;</p>

	прямоугольники)	<p>рассчитать периметр прямоугольника; рассчитать площадь прямоугольника. г. Перегрузить (переопределить): сравнение(меньше); сложение(бинарное).</p>
4	<p>Выполнить построение объектной модели следующей предметной области: “система расчета выгодной покупки”.</p>	<p>Создать класс Money, разработав следующие элементы класса:</p> <p>а. Поля: int first; //номинал купюры int second; //количество купюр</p> <p>б. Конструктор, позволяющий создать экземпляр класса с заданными значениям полей.</p> <p>в. Методы, позволяющие: вывести номинал и количество купюр; определить, хватит ли денежных средств на покупку товара на сумму рублей. определить, сколько шт товара стоимости рублей можно купить на имеющиеся денежные средства.</p> <p>г. Перегрузить (переопределить): сложение(унарное, постфиксное); сложение, совмещённое с присваиванием;</p>
5	<p>Выполнить построение объектной модели следующей предметной области: “диспетчер задач”, при чем выделить объект массив, который будет содержать “handler” процесса.</p>	<p>Создать класс для работы с одномерным массивом целых чисел. Разработать следующие элементы класса:</p> <p>а. Поля: int *ptr; int n.</p> <p>б. Конструктор, позволяющий создать массив размерности n:</p> <p>в. Методы, позволяющие: ввести элементы массива с клавиатуры; вывести элементы массива на экран; отсортировать элементы массива в порядке возрастания.</p> <p>г. Перегрузить (переопределить):</p>

		<p>вычитание, совмещённое с присваиванием</p> <p>логическое отрицание, НЕ</p>
6	<p>Выполнить построение объектной модели следующей предметной области:</p> <p>Игра “судoku”.</p>	<p>Создать класс для работы с двумерным массивом целых чисел. Разработать следующие элементы класса:</p> <p>а. Поля:</p> <pre>int **ptr; int n, m.</pre> <p>б. Конструктор, позволяющий создать массив размерности $n \times n$.</p> <p>в. Методы, позволяющие:</p> <ul style="list-style-type: none"> ввести элементы массива с клавиатуры; вывести элементы массива на экран; вычислить сумму элементов i-того столбца. <p>г. Перегрузить (переопределить):</p> <ul style="list-style-type: none"> равенство присвоение
7	<p>Выполнить построение объектной модели следующей предметной области:</p> <p>“графический редактор простых фигур”, при чем учесть что все графические преобразования выполняются в матричном представлении.</p>	<p>Создать класс для работы с двумерным массивом вещественных чисел. Разработать следующие элементы класса:</p> <p>а. Поля:</p> <pre>double **ptr; int n, m.</pre> <p>б. Конструктор, позволяющий создать массив размерности $n \times m$.</p> <p>в. Методы, позволяющие:</p> <ul style="list-style-type: none"> ввести элементы массива с клавиатуры; вывести элементы массива на экран; отсортировать элементы каждой строки массива в порядке убывания. <p>г. Перегрузить (переопределить):</p> <ul style="list-style-type: none"> умножение обращение к элементу массива

8	Выполнить построение объектной модели следующей предметной области: “анализатор текста, нахождение плагиата”	Создать класс для работы со строками. Разработать следующие элементы класса: а. Поля: char *str; int n. б. Конструктор, позволяющий создать строку из символов. в. Методы, позволяющие: подсчитать количество пробелов в строке; заменить в строке все прописные символы на строчные; удалить из строки все знаки препинания. г. Перегрузить (переопределить): деление операция “логический сдвиг вправо”
9	Выполнить построение объектной модели следующей предметной области: “редактор текстовых файлов”	Создать класс для работы с регулярными выражениями. Разработать следующие элементы класса: а. Поля: Regex r; string text; б. Методы, позволяющие: определить, содержит ли текст фрагменты, соответствующие шаблону поля; вывести на экран все фрагменты текста, соответствующие шаблону поля; удалить из текста все фрагменты, соответствующие шаблону поля; г. Перегрузить (переопределить): умножение, совмещённое с присваиванием операция “модуль”
10	Выполнить построение объектной модели следующей предметной области:	Создать класс для работы с датой. Разработать следующие элементы класса: а. Поле DateTime data.

	“органайзер”	б. Конструкторы, позволяющие установить: заданную дату; дату 1.01.2009. в. Методы, позволяющие: вычислить дату предыдущего дня; вычислить дату следующего дня; определить сколько дней осталось до конца месяца. г. Перегрузить (переопределить): логический сдвиг влево операция “больше или равно”
--	--------------	---

Вариант 8

Задание 1. Выполнить построение объектной модели следующей предметной области: “анализатор текста, нахождение плагиата”

Задание 2.

Создать класс для работы со строками. Разработать следующие элементы класса:

- а. Поля:
 char *str;
 int n.
- б. Конструктор, позволяющий создать строку из символов.
- в. Методы, позволяющие:
 подсчитать количество пробелов в строке;
 заменить в строке все прописные символы на строчные;
 удалить из строки все знаки препинания.
- г. Перегрузить (переопределить):
 деление
 операция “логический сдвиг вправо”

```
#include <iostream>
#include <memory>
#include <cstring>
#include <algorithm>
```



```

template <typename Alloc = std::allocator<char>>
class String_ {
    using alloc_traits = std::allocator_traits<Alloc>;

private:
    char* str;
    size_t length;
    Alloc allocator_;

    void free() {
        if (str) {
            allocator_.deallocate(str, length + 1);
            str = nullptr;
            length = 0;
        }
    }

    void reallocate(size_t new_size) {
        char* new_str = allocator_.allocate(new_size + 1);
        std::memcpy(new_str, str, length);
        free();

        str = new_str;
        length = new_size;
    }

    void copyChars(const char* src, size_t n) {
        reallocate(n);
        std::memcpy(str, src, n);
        str[n] = '\0';
    }

    void appendChars(const char* src, size_t n) {
        reallocate(length + n);
        std::memcpy(str + length, src, n);
        length += n;
        str[length] = '\0';
    }

    void appendChar(char c) {
        reallocate(length + 1);
        str[length++] = c;
    }

```

```

        str[length] = '\\0';
    }

    void appendString(const String_& other) { appendChars(other.str,
other.length); }

    void appendCString(const char* src) { appendChars(src,
std::strlen(src)); }

    void insertChars(size_t pos, const char* src, size_t n) {
        reallocate(length + n);
        std::memmove(str + pos + n, str + pos, length - pos);
        std::memcpy(str + pos, src, n);
        length += n;
        str[length] = '\\0';
    }

    void erase(size_t pos, size_t n) {
        if (pos + n > length) n = length - pos;
        std::memmove(str + pos, str + pos + n, length - (pos + n));
        length -= n;
        str[length] = '\\0';
    }

    void replace(size_t pos, size_t n, const char* src, size_t m) {
        erase(pos, n);
        insertChars(pos, src, m);
    }

    inline int compare(const String_& other) const { return
std::strcmp(str, other.str); }

public:
    String_() : str(nullptr), length(0) {}

    String_(const char* s) : str(nullptr), length(0) {
        copyChars(s, std::strlen(s));
    }

    String_(const String_& other) : str(nullptr), length(0),
allocator_(other.allocator_) {
        copyChars(other.str, other.length);
    }

```

```

    String_(String_&& other) noexcept : str(other.str),
length(other.length), allocator_(std::move(other.allocator_)) {
    other.str = nullptr;
    other.length = 0;
}

~String_() { free(); }

    inline size_t count_spaces() const { return std::count(str, str +
length, ' '); }

    inline void to_lower_case() {
        for (size_t i = 0; i < length; ++i) str[i] =
std::tolower(str[i]);
    }

void remove_punctuation_marks() {
    size_t newLength = 0;
    for (size_t i = 0; i < length; ++i) {
        if (!std::ispunct(str[i]))
            str[newLength++] = str[i];
    }

    length = newLength;
    str[length] = '\0';
}

String_ operator/(size_t divisor) const {
    size_t newLength = length / divisor
;
    String_ result;
    result.copyChars(str, newLength);

    return result;
}

String_ operator>>(size_t n) const {
    if (n >= length) return String_();

    String_ result;
    result.copyChars(str + n, length - n);
}

```

```

        return result;
    }

    inline friend std::ostream& operator<<(std::ostream& os, const
String_& str_obj) {
        os << str_obj.str;
        return os;
    }

    inline size_t size() const { return length; }

    inline const char* c_str() const { return str; }
};

int main() {
    String_<> s("Hello, World! ZXC raddan, international winner !!! I AM
KAGGLE EXPERT");
    std::cout << "Строка: " << s << "\n";

    std::cout << "Кол-во пробелов: " << s.count_spaces() << "\n";

    s.to_lower_case();
    std::cout << "Нижний регистр: " << s << "\n";

    s.remove_punctuation_marks();
    std::cout << "Удаление знаков препинания: " << s << "\n";

    String_<> s2 = s / 2;
    std::cout << "Деление длины на 2: " << s2 << "\n";

    String_<> s3 = s >> 3;
    std::cout << "Сдвиг вправо на 3: " << s3 << "\n";

    return 0;
}

```

```
Строка: Hello, World! ZXC raddan, international winner !!! I AM KAGGLE EXPERT
Кол-во пробелов: 10
Нижний регистр: hello, world! zxc raddan, international winner !!! i am kaggle expert
Удаление знаков препинания: hello world zxc raddan international winner i am kaggle expert
Деление длины на 2: hello world zxc raddan internat
Сдвиг вправо на 3: lo world zxc raddan international winner i am kaggle expert
```

Вывод: приобрели практические навыки создания класса на языке C++.