

# Лабораторная работа №7

## «Разработка простейшего класса на C++»

### 1. Цель работы

Целью данной работы является изучение базовых объектно-ориентированных возможностей языка C++.

### 2. Исходные данные

Рассмотрим некоторые возможности языка C++, которые могут понадобиться при выполнении лабораторной работы.

#### 2.1 Перегрузка операции индексации

Во всех заданиях требуется организовать доступ к элементам внутреннего состояния объектов разрабатываемого класса по индексу. В зависимости от задания, такими элементами могут быть элементы матрицы, коэффициенты полинома, точки ломаной линии и т.п.

Индексацию внутреннего состояния объектов класса можно осуществить путём объявления метода, возвращающего ссылку на элемент с указанным индексом. Например, метод `access`, возвращающий ссылку на элемент вектора, может быть объявлен как

```
class Vector {  
    ...  
    int& access (int index);  
    ...  
};
```

Объявив метод `access`, мы можем использовать его для чтения и записи элементов вектора:

```
Vector v;  
v.access(0)=v.access(1);
```

Однако, гораздо естественнее обращаться к элементам вектора с помощью операции индексации. Чтобы эту операцию можно было применять к объектам нашего класса, достаточно переименовать объявленный нами метод `access` в `operator[]`:

```
class Vector {  
    ...  
    int& operator [] ( int index );  
    ...  
};
```

После этого работа с элементами вектора будет выглядеть как

```
Vector v;  
v[0]=v[1];
```

В случае, если обращение к элементам внутреннего состояния объектов требует использования нескольких индексов (например, если объект представляет матрицу),

перегрузка операции индексации вышеописанным методом становится невозможна, т.к. операция индексации имеет только один формальный параметр.

Тем не менее, для решения этой задачи можно применить приём, являющийся частным случаем известного из функционального программирования метода закармливания функции, а именно: перегруженная операция индексации для, например, матрицы должна принимать первый индекс (номер строки матрицы) и возвращать объект класса, представляющий строку матрицы; при этом для строки матрицы операция индексации перегружена таким образом, что она принимает второй индекс (номер столбца) и возвращает ссылку на элемент матрицы.

В коде это может выглядеть следующим образом:

```
class Matrix {  
    ...  
    class Row {  
        private:  
            Matrix * matrix;  
            int i;  
        public:  
            Row(Matrix * matrix, int i);  
            int& operator [] (int j);  
    };  
  
    Matrix :: Row operator [] (int i)  
    {  
        return Row(this , i);  
    }  
    ...  
};
```

В данном примере внутри класса Matrix объявлен вспомогательный класс Row, объект которого запоминает указатель на матрицу и номер строки матрицы. Операция индексации класса Matrix возвращает временный объект класса Row.

В результате обращение к элементам матрицы может выглядеть как

```
Matrix m;  
m[1][2]= m[3][4];
```

## 2.2 Вывод объектов в поток вывода

Чтобы объекты нашего класса можно было передавать в поток вывода (в частности, в cout), следует перегрузить операцию << для сочетания типов операндов, в котором левый операнд – &ostream, а правый – объект нашего класса. Перегрузку этой операции можно осуществить путём определения глобальной функции operator<<, имеющей параметры указанных типов.

Например, для гипотетического класса Matrix эту функцию можно было бы записать следующим образом:

```
std :: ostream& operator<< (std :: ostream& os, Matrix& matr)  
{  
    for(int i = 0; i < matr.m; i++) {  
        for(int j = 0; j < matr.n; j++) {  
            os << matr[i][j] << ' ';  
        }  
    }
```

```
        os << std :: endl;
    }
    return os;
}
```

### 3 Задание

Выполнение лабораторной работы заключается в составлении на языке C++ программы, состоящей из трёх файлов:

- заголовочный файл `declaration.h` с объявлением одного из классов, приведённых в таблицах 1 – 16;
- файл `implementation.cpp` с определениями методов класса;
- файл `main.cpp`, содержащий функцию `main` и, возможно, вспомогательные функции для проверки работоспособности класса.

**Реализация класса не должна опираться на стандартные контейнерные классы C++**, то есть внутреннее состояние объектов класса должно быть реализовано через обычные массивы. Соответственно, в классе обязательно требуется реализовать:

- конструктор копий;
- деструктор (должен быть объявлен виртуальным);
- операцию присваивания.

Проверку работоспособности класса требуется организовать в функции `main`, размещённой в файле `main.cpp`.

Проверка должна включать в себя:

- создание объекта класса в автоматической памяти;
- передачу объекта класса по значению в функцию;
- присваивание объекта класса переменной.

### 4 Дополнительная информация

- Перегрузка операторов в C++ / Хабр <https://habr.com/ru/post/132014/>
- C++. Перегрузка оператора индексирования элементов массива <https://www.bestprog.net/ru/2019/06/04/overloading-operator-indexing-elements-of-the-array-ru/>

- 1 Египетская дробь  
 $[a_0, a_1, \dots, a_n] = (1/a_0) + (1/a_1) + \dots + (1/a_n)$ .
- в которой  $a_i \in \mathbb{N}$ , с операциями:
1. получение количества  $n$  простых дробей в составе египетской дроби;
  2. получение ссылки на  $a_i$  ;
  3. перевод в рациональное число.
- Для представления рациональных чисел требуется реализовать класс нормализованных дробей с необходимыми арифметическими операциями.
- Конструктор египетской дроби принимает в качестве параметра рациональное число, которое нужно представить в виде египетской дроби, и максимальное количество простых дробей в составе египетской дроби. Для построения египетской дроби целесообразно реализовать алгоритм Фибоначчи.
- 2 Система равенств вида  $a = b$ , где как  $a$ , так и  $b$  может быть либо именем переменной, либо целым числом. Для системы должны быть реализованы операции:
1. получение ссылки на  $i$ -тое равенство;
  2. добавление нового равенства;
  3. проверка, существует ли решение системы.
- 3 Квадратная матрица, элементами которой являются строки, с операциями:
1. получение размера матрицы;
  2. получение ссылки на указанный элемент;
  3. определение, является ли указанный элемент одновременно наименьшим в своей строке матрицы и наибольшим в своём столбце.
- 4 Последовательность символов ASCII с операциями:
1. получение количества символов;
  2. получение ссылки на  $i$ -тый символ;
  3. удаление из последовательности латинских гласных (для представления результата удаления должен создаваться новый объект);
  4. проверка, является ли последовательность гипердромом, т.е. можно ли из её символов построить палиндром.
- 5 Полином степени  $n$  с целочисленными коэффициентами и операциями:
1. вычисление значения для заданного  $x$ ;
  2. получение степени полинома;
  3. получение ссылки на указанный коэффициент

полинома;

4. домножение на полином.

- 6 Квадратная матрица, элементы которой являются рациональными числами, с операциями:
1. получение порядка матрицы;
  2. получение ссылки на указанный элемент;
  3. формирование подматрицы, полученной путём вычёркивания  $i$ -той строки и  $j$ -го столбца;
  4. вычисление определителя матрицы.

Для представления рациональных чисел требуется реализовать класс нормализованных дробей с необходимыми арифметическими операциями.

Конструктор матрицы должен принимать в качестве параметра её порядок и формировать нулевую матрицу.

- 7 Множество интервалов вида  $(a, b)$  на вещественной прямой с операциями:
1. получение количества интервалов;
  2. получение ссылки на  $i$ -тый интервал;
  3. проверка принадлежности числа  $x$  множеству вещественных чисел, задаваемому множеством интервалов;
  4. добавление интервалов одного множества в другое множество;

Для представления интервала следует разработать отдельный класс.

- 8 Функция, представляющая собой сумму синусоид:

$$f(x) = \sum (a_i \cdot \sin(d_i + \omega_i \cdot x)).$$

Каждая синусоида определяется тремя параметрами: амплитуда  $a_i$ , смещение  $d_i$  и частота  $\omega_i$ .

В классе должны быть реализованы операции:

1. получение количества синусоид;
2. получение ссылки на  $i$ -тую синусоиду;
3. вычисление значения функции в точке  $x$ ;
4. формирование новой функции, составленной из синусоид текущей функции, амплитуды которых превышают заданное значение  $a$ .

Для представления синусоиды следует разработать отдельный класс.

- 9 Стековая машина, оперирующая вещественными числами, с операциями:

1. получение количества чисел в стеке;
2. добавление вещественного числа на стек;
3. получение ссылки на  $i$ -ое число, считая от вершины стека;
4. сложение, умножение и вычитание двух чисел на вершине стека (числа удаляются со стека, результат добавляется в стек).

- 10 Полином степени  $n$  с целочисленными коэффициентами и операциями:
1. вычисление значения для заданного  $x$ ;
  2. получение степени полинома;
  3. получение ссылки на указанный коэффициент полинома;
  4. сложение с полиномом (для суммы формируется новый полином, степень которого может стать меньше степеней слагаемых).
- 11 Система материальных точек, каждая из которых определяется своими координатами на плоскости и массой, с операциями:
1. получение количества точек в системе;
  2. получение ссылки на  $i$ -тую точку;
  3. вычисление центра масс системы;
  4. добавление новой точки;
  5. удаление точек с нулевой массой;

Для представления материальной точки следует разработать отдельный класс.

- 12 Очередь целых чисел, реализованная через двойной стек, с операциями:
1. получение количества чисел в очереди;
  2. получение ссылки на  $i$ -ое число (последнее добавленное число имеет индекс 0);
  3. добавление числа в очередь;
  4. удаление числа из очереди.
- 13 Траектория движения материальной точки на плоскости, представленная координатами положения точки в дискретные моменты времени (0 с, 1 с, 2 с, ...), с операциями:
1. получение общего времени движения точки;
  2. получение ссылки на положение точки в момент времени  $t$ ;
  3. добавление нового положения точки в конец траектории;
  4. вычисление пройденного расстояния.

Для представления координат точки следует разработать отдельный класс.

- 14 Конечная цепная дробь  
 $[a_0; a_1, a_2, \dots, a_n] = a_0 + 1 / (a_1 + (1/(a_2 + \dots)))$ , в которой  $a_i \in \mathbb{Z}$  и  $\forall i > 0 : a_i > 0$ , с операциями:

1. получение длины  $n$  дроби;
2. получение ссылки на коэффициент  $a_i$  ;
3. добавление нового коэффициента в конец списка коэффициентов;
4. перевод в рациональное число.

Для представления рациональных чисел требуется реализовать класс нормализованных дробей с необходимыми арифметическими операциями.

Конструктор цепной дроби принимает в качестве параметра рациональное число.

- 15 Запись шахматных партий с операциями:
  1. получение количества сделанных ходов;
  2. получение ссылки на  $i$ -тый ход;
  3. добавление хода;
  4. вычисление количества фигур, оставшихся на доске к концу партии.
- 16 Вещественная матрица размером  $m \times n$  с операциями:
  1. получение количества строк и столбцов;
  2. получение ссылки на указанный элемент;
  3. добавление строки снизу или добавление столбца справа (формируется новая матрица на основе текущей).
- 17 Доска для игры в крестики-нолики размером  $n \times n$  с операциями:
  1. получение размера доски;
  2. получение ссылки на клетку с указанными координатами;
  3. определение, является ли текущая позиция финальной.
- 18 Целочисленная матрица размером  $m \times n$  с операциями:
  1. получение количества строк;
  2. получение количества столбцов;
  3. получение ссылки на указанный элемент;
  4. перестановка двух строк;
  5. перестановка двух столбцов.
- 19 Абсолютный путь к каталогу в файловой системе UNIX с операциями:
  1. получение количества каталогов в пути;
  2. получение ссылки на  $i$ -тый каталог в пути, считая от корня;
  3. добавление имени каталога в конец пути;
  4. получение количества файлов в каталоге (для этого требуется разобраться, как получить оглавление каталога).
- 20 Последовательность символов ASCII, часть из которых может быть неизвестна, с операциями:

1. получение количества символов;
2. проверка, известен ли  $i$ -тый символ;
3. получение ссылки на  $i$ -тый символ (если символ – неизвестен, ему присваивается значение ноль);
4. проверка, можно ли подобрать такие значения неизвестных символов, что последовательность станет палиндромом.

Конструктор последовательности должен формировать последовательность размера  $n$ , все символы которой неизвестны.

- 21 Функция, представляющая собой сумму синусоид:

$$f(x) = \sum (a_i \cdot \sin(d_i + \omega_i \cdot x)).$$

Каждая синусоида определяется тремя параметрами: амплитуда  $a_i$ , смещение  $d_i$  и частота  $\omega_i$ . В классе нужно реализовать операции:

1. получение количества синусоид;
2. получение ссылки на  $i$ -тую синусоиду;
3. вычисление значения функции в точке  $x$ ;
4. добавление новой синусоиды;
5. удаление пар синусоид, которые в сумме дают ноль.

Для представления синусоиды следует разработать отдельный класс.

- 22 Последовательность символов ASCII с операциями:

1. получение количества символов;
2. получение ссылки на  $i$ -тый символ;
3. вставка нового символа в  $i$ -тую позицию последовательности;
4. проверка, является ли последовательность палиндромом.

- 23 Беззнаковое рациональное число, представленное в позиционной системе счисления по основанию  $d$ , часть цифр которого может быть неизвестно. Число должно быть представлено вектором цифр, каждая из которых является целым числом от 0 до  $p - 1$ , и позицией точки, разделяющей целую и дробную часть, в этом векторе. Договоримся о нумерации цифр числа: пусть цифры целой части пронумерованы неотрицательными числами (младшая цифра имеет номер 0), а цифры дробной части имеют отрицательные номера (старшая цифра имеет номер  $-1$ ). Обратите внимание на то, что



позиция точки может быть отрицательна, а также может превышать количество цифр в числе. В классе должны быть реализованы следующие операции:

1. получение количества цифр в числе;
2. определение, известна ли  $i$ -тая цифра числа (если  $i$  выходит за границы допустимых номеров цифр, считать, что  $i$ -тая цифра известна и равна 0);
3. получение ссылки на  $i$ -тую цифру числа (если  $i$  выходит за границы допустимых номеров цифр, нужно расширить число нулями до нужного размера; если  $i$ -тая цифра неизвестна, нужно сделать её известной и обнулить);
4. округление числа до указанного количества значащих цифр.

Конструктор разработанного класса должен принимать в качестве параметра основание системы счисления  $d$ , количество цифр в числе  $n$  и позицию точки  $p$ . При этом он должен порождать число, все цифры которого неизвестны.

- 24 Функция, представляющая собой сумму синусоид:

$$f(x) = \sum (a_i \cdot \sin(d_i + \omega_i \cdot x)).$$

Каждая синусоида определяется тремя параметрами: амплитуда  $a_i$ , смещение  $d_i$  и частота  $\omega_i$ . В классе нужно реализовать операции:

1. получение количества синусоид;
2. получение ссылки на  $i$ -тую синусоиду;
3. вычисление значения функции в точке  $x$ ;
4. добавление синусоид одной функции к другой функции;

Для представления синусоиды следует разработать отдельный класс.

- 25 Ломаная линия на плоскости с операциями:

1. получение количества точек;
2. получение ссылки на  $i$ -тую точку;
3. конкатенация двух ломаных (формируется новая ломаная);
4. выделение части ломаной с  $i$ -той по  $j$ -тую точки (формируется новая ломаная).

Точки на плоскости должны быть представлены структурами с целочисленными полями  $x$  и  $y$ .

- 26 Вещественная матрица размером  $m \times n$  с операциями:
1. получение количества строк;
  2. получение количества столбцов;
  3. получение ссылки на указанный элемент;
  4. умножение на число;
  5. транспонирование (формируется новая матрица на основе текущей).
- 27 Очередь целых чисел, реализованная через кольцевой буфер, с операциями:
1. получение количества чисел в очереди;
  2. получение ссылки на  $i$ -ое число (головной элемент имеет индекс 0);
  3. добавление числа в хвост очереди;
  4. удаление головного элемента очереди.
- 28 Полином степени  $n$  с вещественными коэффициентами и операциями:
1. вычисление значения для заданного  $x$ ;
  2. получение степени полинома;
  3. получение ссылки на указанный коэффициент полинома;
  4. вычисление производной (формируется новый полином).
- 29 Конечная цепная дробь  $[a_0; a_1, a_2, \dots, a_n] = a_0 + 1 / (a_1 + (1/(a_2 + \dots)))$ , в которой  $a_i \in \mathbb{Z}$  и  $\forall i = 1, 2, 3, \dots, n : (a_i > 0)$ , с операциями:
1. получение длины  $n$  дроби;
  2. сокращение длины дроби до указанного  $n_0$ ;
  3. получение ссылки на  $a_i$ ;
  4. перевод в рациональное число.

Для представления рациональных чисел требуется реализовать класс нормализованных дробей с необходимыми арифметическими операциями.

Конструктор цепной дроби принимает в качестве параметра вещественное число  $x$  и точность  $\epsilon$ .

Длина цепной дроби должна быть выбрана таким образом, чтобы представляемое ею число  $x'$  удовлетворяло неравенству:

$$|(x' - x) / x| < \epsilon.$$

- 30 Траектория движения материальной точки на плоскости, представленная координатами положения точки в дискретные моменты

времени (0 с, 1 с, 2 с, ...), с операциями:

1. получение общего времени движения точки;
2. получение ссылки на положение точки в момент времени  $t$ ;
3. добавление одной траектории в конец другой траектории;

Для представления координат точки следует разработать отдельный класс.

- 31 Последовательность строк, имеющая переменную длину, с операциями:

1. получение длины последовательности;
2. получение ссылки на  $i$ -тую строку;
3. вставка строки в указанное место последовательности;
4. удаление из последовательности дублирующихся строк (если есть две или более одинаковые строки, остаётся только первая).

Для представления строк нужно использовать стандартный класс `string`.

- 32 Матрица смежности для простого графа с операциями:

1. определение, смежны ли две вершины;
2. добавление вершины;
3. удаление вершины;
4. вычисление количества компонент связности графа.

- 33 Полином степени  $n$  с вещественными коэффициентами и операциями:

1. получение степени полинома;
2. получение ссылки на указанный коэффициент полинома;
3. вычисление первообразной (формируется новый полином, значение младшего коэффициента первообразной передаётся в качестве параметра).

- 34 Квадратная матрица, элементами которой являются упорядоченные множества строк, с операциями:

1. получение ссылки на элемент  $(i, j)$ ;
2. умножение матрицы на саму себя.

При выполнении умножения матрицы считать, что для множеств строк операции сложения и

умножения определены следующим образом:

**сложение множеств  $A$  и  $B$**  – объединение этих множеств;

**умножение множеств  $A$  и  $B$**  – результатом умножения является множество, составленное из конкатенаций всех возможных пар строк таких, что первая строка принадлежит  $A$ , а вторая – принадлежит  $B$ .

Для представления строк нужно использовать стандартный класс `string`.

- 35 Матрица смежности для простого орграфа с операциями:
1. определение, ведёт ли дуга из  $i$ -той вершины в  $j$ -тую (операция индексации);
  2. добавление вершины;
  3. удаление вершины;
  4. определение, содержит ли граф циклы.