

Лабораторная работа №9

«Разработка простейшего класса на C++»

Скоробогатов С.Ю.

26 апреля 2016 г.

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 матрицы, может быть объявлен как

```
class Matrix {  
    ...  
    int & access(int i, int j);  
    ...  
};
```

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

```
Matrix m;  
m.access(0,0) = m.access(1,1);
```

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

```
class Matrix {  
    ...  
    int & operator[] (int i, int j);  
    ...  
};
```

После этого работа с элементами матрицы будет выглядеть как

```
Matrix m;  
m[0,0] = m[1,1];
```

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` эта функция может иметь следующий вид:

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

3 Задание

Согласно выбранному из таблиц 1–10 описанию требуется составить два варианта класса:

- в варианте «А» в классе объявляется поле типа `vector` для хранения элементов матрицы, коэффициентов полинома, точек ломаной линии и т.п.
- в варианте «Б» хранение вышеперечисленных данных следует организовать с помощью обычных массивов.

В варианте «А» класса не требуется объявлять деструктор и копирующий конструктор, и не нужно перегружать операцию присваивания. В варианте «Б» наличие «большой тройки» обязательно.

Для обоих вариантов объявление класса следует разместить в заголовочном файле, а определения его методов – в отдельном файле с расширением `«.cpp»`.

Проверку работоспособности класса требуется организовать в функции `main`, размещённой в файле `«main.cpp»`. Проверка должна включать в себя:

1. создание объекта класса в автоматической памяти;
2. передачу объекта класса по значению в функцию;
3. присваивание объекта класса переменной.

Для проверки обоих вариантов класса должен использоваться один и тот же файл `«main.cpp»`.

Таблица 1: Варианты заданий

1	<p>Целочисленная матрица размером $m \times n$ с операциями:</p> <ol style="list-style-type: none"> 1. получение количества строк; 2. получение количества столбцов; 3. получение ссылки на указанный элемент; 4. удаление строки (формируется новая матрица на основе текущей); 5. удаление столбца (формируется новая матрица на основе текущей).
2	<p>Полином степени n с вещественными коэффициентами и операциями:</p> <ol style="list-style-type: none"> 1. вычисление значения для заданного x; 2. получение степени полинома; 3. получение ссылки на указанный коэффициент полинома; 4. вычисление производной (формируется новый полином).
3	<p>Ломаная линия на плоскости с операциями:</p> <ol style="list-style-type: none"> 1. получение количества точек; 2. вычисление длины ломаной; 3. получение ссылки на i-тую точку; 4. добавление точки в конец ломаной. <p>Точки на плоскости должны быть представлены структурами с вещественными полями x и y.</p>
4	<p>Конечная цепная дробь</p> $[a_0; a_1, a_2, \dots, a_n] = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \dots}}$ <p>в которой $a_i \in \mathbb{Z}$ и $\forall i = \overline{1, n} : (a_i > 0)$,</p> <p>с операциями:</p> <ol style="list-style-type: none"> 1. получение длины n дроби; 2. получение ссылки на a_i; 3. перевод в рациональное число. <p>Для представления рациональных чисел требуется реализовать класс нормализованных дробей с необходимыми арифметическими операциями. Конструктор цепной дроби принимает в качестве параметра рациональное число.</p>

Таблица 2: Варианты заданий

5	<p>Вещественная матрица размером $m \times n$ с операциями:</p> <ol style="list-style-type: none"> 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).
6	<p>Полином степени n с вещественными коэффициентами и операциями:</p> <ol style="list-style-type: none"> 1. вычисление значения для заданного x; 2. получение степени полинома; 3. получение ссылки на указанный коэффициент полинома; 4. вычисление первообразной (формируется новый полином, значение младшего коэффициента первообразной передаётся в качестве параметра).
7	<p>Ломаная линия на плоскости с операциями:</p> <ol style="list-style-type: none"> 1. получение количества точек; 2. вычисление размеров минимального прямоугольника, внутри которого может поместиться ломаная; 3. получение ссылки на i-тую точку; 4. вставка новой точки в любое место ломаной. <p>Точки на плоскости должны быть представлены структурами с целочисленными полями x и y.</p>
8	<p>Конечная цепная дробь</p> $[a_0; a_1, a_2, \dots, a_n] = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \dots}}$ <p>с операциями:</p> <ol style="list-style-type: none"> 1. получение длины n дроби; 2. получение ссылки на a_i; 3. перевод в число с плавающей точкой. <p>Конструктор цепной дроби принимает в качестве параметра число с плавающей точкой и длину дроби.</p>

Таблица 3: Варианты заданий

9	<p>Целочисленная матрица размером $m \times n$ с операциями:</p> <ol style="list-style-type: none"> 1. получение количества строк; 2. получение количества столбцов; 3. получение ссылки на указанный элемент; 4. перестановка двух строк; 5. перестановка двух столбцов.
10	<p>Полином степени n с целочисленными коэффициентами и операциями:</p> <ol style="list-style-type: none"> 1. вычисление значения для заданного x; 2. получение степени полинома; 3. получение ссылки на указанный коэффициент полинома; 4. домножение на полином.
11	<p>Замкнутая ломаная линия на плоскости с операциями:</p> <ol style="list-style-type: none"> 1. получение количества отрезков, из которых состоит ломаная; 2. получение ссылки на i-тую точку; 3. удаление отрезков, длина которых меньше указанного значения. <p>Точки на плоскости должны быть представлены структурами с вещественными полями x и y.</p>
12	<p>Конечная цепная дробь</p> $[a_0; a_1, a_2, \dots, a_n] = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \dots}}$ <p>в которой $a_i \in \mathbb{Z}$ и $\forall i = \overline{1, n} : (a_i > 0)$, с операциями:</p> <ol style="list-style-type: none"> 1. получение длины n дроби; 2. сокращение длины дроби до указанного значения; 3. получение ссылки на a_i; 4. перевод в рациональное число. <p>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 ε. Длина цепной дроби должна быть выбрана таким образом, чтобы представляемое ею число \tilde{x} удовлетворяло неравенству: $\left \frac{\tilde{x} - x}{x} \right < \varepsilon$.</p>

Таблица 4: Варианты заданий

13	<p>Вещественная матрица размером $m \times n$ с операциями:</p> <ol style="list-style-type: none"> 1. получение количества строк; 2. получение количества столбцов; 3. получение ссылки на указанный элемент; 4. добавление строки снизу; 5. добавление столбца справа (формируется новая матрица на основе текущей).
14	<p>Полином степени n с целочисленными коэффициентами и операциями:</p> <ol style="list-style-type: none"> 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).
15	<p>Замкнутая ломаная линия на плоскости с операциями:</p> <ol style="list-style-type: none"> 1. получение количества точек; 2. получение ссылки на i-тую точку; 3. разбиение отрезков ломаной, длина которых превышает указанное значение, напополам. <p>Точки на плоскости должны быть представлены структурами с вещественными полями x и y.</p>
16	<p>Египетская дробь $[a_0, a_1, \dots, a_n] = \frac{1}{a_0} + \frac{1}{a_1} + \dots + \frac{1}{a_n}$, в которой $a_i \in \mathbb{N}$, с операциями:</p> <ol style="list-style-type: none"> 1. получение количества n простых дробей в составе египетской дроби; 2. получение ссылки на a_i; 3. перевод в рациональное число. <p>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.</p>

Таблица 5: Варианты заданий

17	<p>Матрица рациональных чисел размером $m \times n$ с операциями:</p> <ol style="list-style-type: none"> 1. получение количества строк; 2. получение количества столбцов; 3. получение ссылки на указанный элемент; 4. умножение строки на рациональное число; 5. прибавление одной строки к другой. <p>Для представления рациональных чисел требуется реализовать класс нормализованных дробей с операциями сложения и умножения.</p>
18	<p>Полином степени n с рациональными коэффициентами и операциями:</p> <ol style="list-style-type: none"> 1. вычисление значения для заданного x; 2. получение степени полинома; 3. получение ссылки на указанный коэффициент полинома; 4. деление на полином (для хранения результата формируется новый полином, остаток от деления отбрасывается). <p>Дополнительно требуется реализовать класс нормализованных дробей с операциями сложения и умножения.</p>
19	<p>Ломаная линия на плоскости с операциями:</p> <ol style="list-style-type: none"> 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). <p>Точки на плоскости должны быть представлены структурами с целочисленными полями x и y.</p>
20	<p>Беззнаковое целое число, записанное в системе счисления по основанию 2^{16}, с операциями:</p> <ol style="list-style-type: none"> 1. получение количества цифр; 2. получение ссылки на i-тую цифру (младшая цифра имеет индекс 0); 3. прибавление числа; 4. конкатенация двух чисел (для хранения результата формируется новый объект). <p>Конструктор класса должен принимать начальное значение типа <code>long</code>. Вывод чисел можно выполнять в шестнадцатеричной системе.</p>

Таблица 6: Варианты заданий

21	<p>Вектор значений, каждое из которых является либо целым числом, либо «не определено». Предполагается, что при создании вектора все его элементы не определены. Они становятся определены только после присваивания им целочисленного значения.</p> <p>Для вектора должны быть реализованы следующие операции:</p> <ol style="list-style-type: none"> 1. проверка, определён ли элемент вектора с индексом i; 2. получение ссылки на i-тый элемент (если элемент не определён, ему автоматически присваивается нулевое значение); 3. добавление нового неопределённого элемента в конец вектора. <p>Конструктор вектора должен принимать в качестве параметра его начальный размер.</p>
22	<p>Полином с целочисленными коэффициентами, часть из которых может быть «неизвестна». Для полинома должны быть реализованы следующие операции:</p> <ol style="list-style-type: none"> 1. проверка, определён ли коэффициент полинома с индексом i; 2. получение ссылки на i-тый коэффициент полинома (при этом неизвестному коэффициенту автоматически присваивается нулевое значение); 3. дифференцирование с порождением нового полинома для хранения производной (коэффициенты производной, для вычисления которых нужно знать значения неизвестных коэффициентов полинома, становятся неизвестными).
23	<p>Целочисленная квадратная матрица, часть элементов которой может иметь «неопределённое» значение. Предполагается, что при создании матрицы все её элементы не определены. Они становятся определены только после присваивания им целочисленного значения.</p> <p>Для матрицы должны быть реализованы следующие операции:</p> <ol style="list-style-type: none"> 1. проверка, определён ли элемент с индексами (i, j); 2. запись целого числа в элемент с индексами (i, j); 3. чтение значения элемента с индексами (i, j) (чтение неопределённого элемента автоматически присваивает ему нулевое значение); 4. домножение матрицы на себя. <p>Конструктор матрицы должен принимать в качестве параметра её размер.</p>

Таблица 7: Варианты заданий

24	<p>Целочисленная матрица размера $m \times n$, часть элементов которой может иметь «неопределённое» значение. Предполагается, что при создании матрицы все её элементы не определены. Они становятся определены только после присваивания им целочисленного значения.</p> <p>Для матрицы должны быть реализованы следующие операции:</p> <ol style="list-style-type: none"> 1. проверка, определён ли элемент с индексами (i, j); 2. запись целого числа в элемент с индексами (i, j); 3. чтение значения элемента с индексами (i, j) (чтение неопределённого элемента автоматически присваивает ему нулевое значение); 4. умножение строки матрицы на число; 5. прибавление одной строки матрицы к другой строке. <p>Конструктор матрицы должен принимать в качестве параметров её размеры m и n.</p>
25	<p>Последовательность булевских значений длины n, часть элементов которой может быть «неизвестна». Предполагается, что при создании последовательности все её элементы неизвестны. Они становятся определены только после присваивания им булевского значения.</p> <p>Для последовательности должны быть реализованы следующие операции:</p> <ol style="list-style-type: none"> 1. проверка, известен ли i-тый элемент последовательности; 2. получение ссылки на i-тый элемент последовательности (при этом неизвестному элементу автоматически присваивается значение «false»); 3. конкатенация двух последовательностей; 4. логическое И двух последовательностей равной длины. <p>Конструктор последовательности должен принимать в качестве параметра её длину.</p>
26	<p>Последовательность знаковых 64-битовых чисел с фиксированной точкой, в которой каждое число $x = whole + frac/2^{32}$, где $whole$ – это целая часть (32-битовое целое число со знаком), $frac$ – дробная часть (32-битовое целое число без знака).</p> <p>Для последовательности должны быть реализованы следующие операции:</p> <ol style="list-style-type: none"> 1. получение ссылки на целую часть i-того элемента последовательности; 2. получение ссылки на дробную часть i-того элемента последовательности; 3. умножение каждого числа последовательности на целое 32-разрядное число. <p>Конструктор последовательности должен принимать в качестве параметра её длину.</p>

Таблица 8: Варианты заданий

27	<p>Последовательность строк переменной длины с операциями:</p> <ol style="list-style-type: none"> 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).
28	<p>Квадратная матрица, элементами которой являются упорядоченные множества строк, с операциями:</p> <ol style="list-style-type: none"> 1. получение ссылки на элемент с индексами (i, j); 2. умножение матрицы на саму себя. <p>При выполнении умножения матрицы считать, что для множеств строк операции сложения и умножения определены следующим образом:</p> <p>сложение множеств A и B – объединение этих множеств;</p> <p>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.</p> <p>Конструктор матрицы должен принимать в качестве параметра её размер. Для представления множеств строк можно использовать <code>set<string></code>.</p>
29	<p>Вектор простых дробей с операциями:</p> <ol style="list-style-type: none"> 1. получение размера вектора; 2. получение ссылки на i-тый элемент; 3. вставка нового элемента в указанное место вектора; 4. вычисление скалярного произведения вектора на самого себя. <p>Дополнительно требуется реализовать класс простых дробей с операциями сложения и умножения.</p>

Таблица 9: Варианты заданий

30	<p>Матрица смежности для простого графа с операциями:</p> <ol style="list-style-type: none"> 1. определение, смежны ли две вершины; 2. добавление ребра; 3. удаление ребра; 4. добавление вершины; 5. удаление вершины; 6. вычисление количества компонент связности графа.
31	<p>Матрица смежности для простого орграфа с операциями:</p> <ol style="list-style-type: none"> 1. определение, ведёт ли дуга из i-той вершины в j-тую; 2. добавление дуги; 3. удаление дуги; 4. добавление вершины; 5. удаление вершины; 6. определение, содержит ли граф циклы.
32	<p>Матрица смежности для неориентированного мультиграфа, рёбра которого помечены строками, с операциями:</p> <ol style="list-style-type: none"> 1. определение, смежны ли две вершины; 2. добавление ребра; 3. удаление ребра; 4. добавление вершины; 5. удаление вершины; 6. вычисление множества меток рёбер, инцидентных указанной вершине. <p>Для представления множества строк можно использовать <code>set<string></code>.</p>

Таблица 10: Варианты заданий

33	<p>Отношение R на множестве целых чисел из диапазона $[a, b]$, представленное квадратной матрицей A булевских значений размера $b - a + 1$. Строки и столбцы матрицы соответствуют числам из диапазона $[a, b]$. Подразумевается, что если $A_{i,j}$ – истина, то $(i + a, j + a) \in R$.</p> <p>Для отношения должны быть реализованы следующие операции:</p> <ol style="list-style-type: none"> 1. получение ссылки на элемент матрицы, соответствующий паре чисел (x, y), в которой $a \leq x \leq b, a \leq y \leq b$; 2. определение, является ли отношение рефлексивным; 3. определение, является ли отношение симметричным; 4. построение транзитивного замыкания отношения (при этом должно порождаться новое отношение).
34	<p>Перестановка P порядка n (т.е. биекция множества натуральных чисел от 0 до $n - 1$ на себя) с операциями:</p> <ol style="list-style-type: none"> 1. получение $P(i)$, где $i = \overline{0, n - 1}$; 2. обмен $(P(i) \leftrightarrow P(j))$; 3. вычисление композиции данной перестановки с некоторой перестановкой Q того же порядка; 4. определение, является ли перестановка обратной самой себе. <p>Конструктор перестановки должен получать в качестве параметра её порядок n и создавать тождественную перестановку.</p>
35	<p>Целое беззнаковое число, представленное в позиционной системе счисления по основанию d. Другими словами, число представлено вектором цифр, каждая из которых является целым числом от 0 до $n - 1$. Некоторые цифры числа могут быть неизвестны.</p> <p>В классе должны быть реализованы следующие операции:</p> <ol style="list-style-type: none"> 1. получение количества цифр в числе; 2. определение, известна ли i-тая цифра числа (если i превышает количество цифр в числе, считать, что i-тая цифра известна и равна 0); 3. получение ссылки на i-тую цифру числа (если i превышает количество цифр в числе, нужно расширить число нулями до нужного размера; если i-тая цифра неизвестна, нужно сделать её известной и обнулить); 4. прибавление другого числа, представленного в системе с тем же основанием. <p>Конструктор разработанного класса должен принимать в качестве параметра основание системы счисления d и количество цифр в числе n. При этом он должен порождать число, все цифры которого неизвестны.</p>

Таблица 11: Варианты заданий

36	<p>Беззнаковое рациональное число, представленное в позиционной системе счисления по основанию d. Другими словами, число представлено вектором цифр, каждая из которых является целым числом от 0 до $n - 1$, и позицией точки, разделяющей целую и дробную часть, в этом векторе. Некоторые цифры числа могут быть неизвестны. 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).</p> <p>Обратите внимание на то, что позиция точки может быть отрицательна, а также может превышать количество цифр в числе.</p> <p>В классе должны быть реализованы следующие операции:</p> <ol style="list-style-type: none"> 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. <p>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.</p>
37	
38	