# Лабораторная работа №2

## Цель работы

Освоение методов использования динамической памяти, изучение свойства полиморфизма, реализуемого перегрузкой функций и операций в классах C++.

## Задание к лабораторной работе

Составить описание класса для объектов прямоугольных матриц, задаваемых массивом вещественных чисел типа double, располагающегося в памяти по строкам. Компоненты матрицы должны быть скрыты (инкапсулированы) в объекте.

Предусмотреть применение конструкторов:

а) по умолчанию;

б) для инициализации квадратной матрицы заданного размера с заданными компонентами;

в) для инициализации прямоугольной матрицы заданных размеров с заданными компонентами;

г) копирования.

Можно использовать параметры по умолчанию для сокращения количества конструкторов.

Конструкторы должны создавать объекты в динамической памяти (оператор new), а деструктор – освобождать память (оператор delete). Способ размещения объекта в динамической памяти (в виде одномерного или двумерного динамического массива, либо комбинированный вариант – одномерный массив с массивом указателей на начало каждой строки матрицы) выбрать самостоятельно. Все эти способы имеют как достоинства, так и недостатки.

Организовать в конструкторах и деструкторе вывод на экран информационных сообщений, например, «Конструктор матрицы ХХХ», «Деструктор матрицы ХХХ» и т.д. Вместо «ХХХ» указывать некоторый уникальный идентификатор матрицы.

I. С помощью функций-элементов класса обеспечить:

1) проверку возможности умножения двух матриц;

2) проверку возможности сложения двух матриц;

3) поиск максимального элемента матрицы;

4) поиск минимального элемента матрицы.

II. С помощью операторов-элементов класса обеспечить:

1) доступ к элементам матрицы по индексу строки и столбца (чтение/запись), т.е. переопределить оператор [];

2) математические действия над матрицами A и B без получения новых матриц, т.е. переопределить операторы

а) A = B;

б) A += B;

в) A –= B;

г) A \*= B;

д) а также умножение матрицы на скаляр *k*.

III. C помощью внешних операторов обеспечить:

1) двуместные операции над матрицами *A* и *B* с получением новой матрицы *C*:

а) сложение (C = A + B);

б) вычитание (C = A – B);

в) произведение (С = A \* B);

г) а также умножение матрицы на скаляр *k*.

2) вывод на экран матрицы в построчной форме, т.е. переопределить оператор вывода на поток (<<).

Выполнению операций сложения, вычитания и умножения матриц должна предшествовать проверка возможности их выполнения над данными объектами.

УКАЗАНИЕ: Для выравнивания позиций при выводе матрицы на экран можно использовать функции-манипуляторы потока (библиотека iomanip) либо методы класса ostream (библиотека iostream):

|  |  |
| --- | --- |
| Манипуляторы потока | Методы класса ostream |
| fixed  scientific  setprecision(int n)  setw(int n)  resetiosflags(ios\_base::fmtflags flag)  setiosflags(ios\_base::fmtflags flag) | setf(ios\_base::fixed, ios\_base::floatfield)  setf(ios\_base::scientific, ios\_base::floatfield)  precision(int n)  width(int n)  setf(0, ios\_base::fmtflags flag)  setf(ios\_base::fmtflags flag) |