**Задачи для подготовки к зачету по дисциплине «Программирование на С++».**

**№1.**

1. Написать класс, реализующий динамический массив вещественных элементов. Обеспечить безопасность работы с динамической памятью. Написать метод для создания и ввода массива с клавиатуры. Написать константный метод для вывода массива на экран. Копирование, присваивание, перемещение для объектов этого класса запретить.

2. Для класса из п.1 реализовать перегрузку оператора « **+=** » таким образом, чтобы к каждому элементу массива прибавлялось число (как вещественное, так и целочисленное).

3. Для класса из п.1 реализовать возможность перемещения.

Предусмотреть дружественный интерфейс.

Запрещается использование библиотек, за исключением <iostream> .

Предполагаемое время выполнения: 60 мин.

**№2.**

1. Написать класс, реализующий три динамических массива вещественных элементов. Обеспечить безопасность работы с динамической памятью. Написать метод для создания и ввода массивов с клавиатуры. Написать константный метод для вывода массивов на экран. Копирование, присваивание, перемещение для объектов этого класса запретить.

2. Для класса из п.1 реализовать перегрузку оператора « ( ) » таким образом, чтобы метод принимал два параметра. Первым параметром должен быть номер массива (одного из трех). Вторым параметром должен быть номер элемента в соответствующем массиве.

3. Для класса из п.1 реализовать возможность перемещения.

Предусмотреть дружественный интерфейс.

Запрещается использование библиотек, за исключением <iostream> .

Предполагаемое время выполнения: 60 мин.

**№3.**

1. Написать класс, реализующий двумерный динамический массив вещественных элементов. Обеспечить безопасность работы с динамической памятью. Написать метод для создания и ввода массива с клавиатуры. Написать константный метод для вывода массива на экран. Копирование, присваивание, перемещение для объектов этого класса запретить.

2. Для класса из п.1 реализовать возможность копирования, присваивания, перемещения.

3. Для класса из п.1 реализовать перегрузку оператора сложения. Сложение должно объединять элементы двух матриц в одну матрицу – новую. Размеры (высота и ширина) новой (создаваемой) матрицы должны подбираться максимально оптимально, но они не должны быть меньше минимального размера складываемых матриц. Свободные места в новой матрице можно заполнить нулями.

Предусмотреть дружественный интерфейс.

Запрещается использование библиотек, за исключением <iostream> .

Предполагаемое время выполнения: 80 мин.