

Реализовать соответствующий класс (без использования библиотеки STL). Реализовать в этом классе требуемые в задании операции. Память под члены-данные класса должна выделяться динамически (где это необходимо).

#### Вариант 1.

В классе «**Отображение**» – **Map** перегрузить операции для добавления нового элемента в отображение (сложение), удаления элемента по ключу (вычитание), объединения (сложение), пересечения (умножения), индексирования, присваивания и вывода на экран. Создать массив объектов и передавать пары объектов в функцию, которая строит отображение, состоящее из элементов, входящих только в одно из заданных отображений, т. е.  $(A \cup B) \setminus (A \cap B)$ , и возвращает его в основную программу (использовать при этом реализованные перегруженные операции).

#### Вариант 2.

В классе «**Вектор**» – **Vector** перегрузить операции сложения, вычитания, умножения векторов, умножения на константу, инкремента, декремента, индексирования, присваивания и вывода на экран. Создать массив объектов. Написать функцию, которая для заданной пары векторов будет определять, являются ли они коллинеарными или ортогональными, используя реализованные операции.

#### Вариант 3.

В классе «**Строка**» – **String** перегрузить операции сложения, индексирования, отношения, конкатенации, сравнения, присваивания и вывода на экран. Создать массив объектов и передать его в функцию, которая выполняет сортировку строк, используя реализованные операции.

#### Вариант 4.

В классе «**Мультимножество целых чисел**» – **Multiset** перегрузить операции сложения, вычитания, умножения (пересечения), индексирования, присваивания и вывода на экран. Создать массив объектов и передавать пары объектов в функцию, которая строит мультимножество, состоящее из элементов, входящих только в одно из заданных множеств, т. е.  $(A \cup B) \setminus (A \cap B)$ , и возвращает его в основную программу (использовать при этом реализованные перегруженные операции).

#### Вариант 5.

В классе «**Множество строк**» – **Set** перегрузить операции сложения, вычитания, умножения (пересечения), индексирования, присваивания и вывода на экран. Создать массив объектов и передавать пары объектов в функцию, которая строит множество, состоящее из элементов, входящих только в одно из заданных множеств, т. е.  $(A \cup B) \setminus (A \cap B)$ , и возвращает его в основную программу (использовать при этом реализованные перегруженные операции).

### Вариант 6.

В классе «**Массив строк**» перегрузить операции сложения, умножения, индексирования, присваивания и вывода на экран для данного класса. Создать массив объектов и передавать объекты в функцию, которая выполняет слияние объектов и для полученного объекта-результата производит лексикографическое упорядочения строк (использовать при этом реализованные перегруженные операции).

### Вариант 7.

В классе «**Множество целых чисел**» – **Set** перегрузить операции сложения, вычитания, умножения (пересечения), индексирования, присваивания и вывода на экран. Создать массив объектов и передавать пары объектов в функцию, которая строит множество, состоящее из элементов, входящих только в одно из заданных множеств, т. е.  $(A \cup B) \setminus (A \cap B)$ , и возвращает его в основную программу (использовать при этом реализованные перегруженные операции).

### Вариант 8.

В классе «**Очередь**» – **Queue** реализовать добавление элементов в очередь, извлечение элементов, присваивание и вывод на экран с помощью перегруженных операций. Элементы очереди хранятся в массиве. Если массив имеет фиксированную размерность, то предусмотреть контроль выхода за пределы массива. Если память выделяется динамически и ее не хватает, то увеличить размер выделенной памяти. Создать массив объектов. Передавать объекты в функцию, которая удаляет из очереди первый (с головы очереди), третий, пятый и т. д. (нечётные) элементы (использовать при этом реализованные перегруженные операции).

### Вариант 9.

В классе «**Дробь**» – **Fraction** перегрузить операции сложения, вычитания, умножения, деления, присваивания, операции отношения и вывода на экран. Создать массив объектов и передать его в функцию, которая изменяет каждый элемент массива с четным индексом путем добавления следующего за ним элемента массива, используя реализованные операции.

### Вариант 10.

В классе «**Многочлен**» – **Polynom** степени  $n$  перегрузить операции сложения, вычитания, умножения, инкремента, декремента, индексирования, присваивания, вывода на экран. Создать массив объектов класса. Передать его в функцию, вычисляющую сумму полиномов массива и возвращающую полином-результат, который выводится на экран в основной программе (использовать при этом реализованные перегруженные операции).

### Вариант 11.

В классе «**Стек целых чисел**» – **Stack** реализовать включение элементов в стек, извлечение элементов, присваивание и вывод на экран с помощью перегруженных операций. Элементы стека хранятся в массиве. Если массив имеет фиксированную

размерность, то предусмотреть контроль выхода за пределы массива. Если память выделяется динамически и ее не хватает, то увеличить размер выделенной памяти. Создать массив объектов. Передавать объекты в функцию, которая удаляет из стека первый (сверху), третий, пятый и т. д. (нечётные) элементы (использовать при этом реализованные перегруженные операции).

#### **Вариант 12.**

В классе «**Бинарное дерево целых чисел**» – **Tree** реализовать вставку элемента в дерево, удаления элемента по ключу, присваивание и вывод на экран с помощью перегруженных операций. Сформировать дерево, вывести содержимое его узлов в порядке возрастания, определить высоту дерева – максимальное число узлов, принадлежащих пути от корня дерева до любого из его листьев (использовать при этом реализованные перегруженные операции).

#### **Вариант 13.**

В классе «**Односвязный список целых чисел**» – **List** реализовать включение элементов в список, извлечение элементов, присваивание и вывод на экран с помощью перегруженных операций. Элементы списка хранятся в массиве. Если массив имеет фиксированную размерность, то предусмотреть контроль выхода за пределы массива. Если память выделяется динамически и ее не хватает, то увеличить размер выделенной памяти. Создать массив объектов. Передавать объекты в функцию, которая упорядочивает элементы списка по возрастанию (использовать при этом реализованные перегруженные операции).

#### **Вариант 14.**

В классе «**Комплексное число**» – **Complex** перегрузить операции для сложения, вычитания, умножения, деления, присваивания и вывода на экран. Создать два вектора размерности  $n$  из комплексных координат. Передать их в функцию, которая выполняет сложение комплексных векторов, используя реализованные операции.

#### **Вариант 15.**

В классе «**Стек строк**» – **Stack** реализовать включение элементов в стек, извлечение элементов, присваивание и вывод на экран с помощью перегруженных операций. Элементы стека хранятся в массиве. Если массив имеет фиксированную размерность, то предусмотреть контроль выхода за пределы массива. Если память выделяется динамически и ее не хватает, то увеличить размер выделенной памяти. Создать массив объектов. Передавать объекты в функцию, которая удаляет из стека второй (сверху), четвёртый, шестой и т. д. (чётные) элементы (использовать при этом реализованные перегруженные операции).

#### **Вариант 16.**

В классе «**Квадратная матрица**» – **Matrix** перегрузить операции сложения, вычитания, умножения, присваивания и вывода на экран. Создать массив объектов класса **Matrix** и

передать его в функцию, которая изменяет  $i$ -ю матрицу путем возведения ее в квадрат, используя реализованные операции. В основной программе вывести результат.

#### **Вариант 17.**

В классе «Двусвязный список целых чисел» – **List** реализовать включение элементов в список, извлечение элементов, присваивание, вывод на экран с помощью перегруженных операций. Элементы списка хранятся в массиве. Если массив имеет фиксированную размерность, то предусмотреть контроль выхода за пределы массива. Если память выделяется динамически и ее не хватает, то увеличить размер выделенной памяти. Создать два упорядоченных по возрастанию списка, слить их в один (также упорядоченный по возрастанию), построив новый список, и вывести результат на экран (использовать при этом реализованные перегруженные операции).