Реализовать в соответствующем шаблонном классе-контейнере **обработку ошибочных ситуаций с помощью аппарата исключений** (try-throw-catch). Необходимо отслеживать следующие ошибки:

- Ошибки работы с файлами (потоками) при считывании исходных данных
- Ошибки работы с динамической памятью
- Выход за границы массивов
- Добавление в уже полный контейнер (если размер контейнера ограничен), извлечение из пустого контейнера.

Также реализовать **журналирование** в методах класса, которые указаны в задании. Требуется выводить в лог-файл:

- Сообщение при входе в функцию (с указанием переданных значений параметров);
- Сообщение при выходе из функции;
- При возникновении исключительной ситуации выводить тип исключения и сообщение об ошибке.

Продемонстрировать функциональность реализованного класса с учётом случаев возникновения различных типов исключительных ситуаций и журналирования вызовов функций работы с контейнером.

Память под члены-данные класса должна выделяться динамически (где это необходимо). Использовать классы библиотеки STL не разрешается.

Вариант 1.

«Стек» – Stack. 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. В классе реализовать методы для включения элементов в стек и их извлечения из стека. Перегрузить операции сложения, вычитания, присваивания и вывода на экран. Создать массив объектов. Передавать объекты в функцию, которая удаляет из стека первый (сверху), третий, пятый и т. д. (нечётные) элементы.

Вариант 2.

«Отображение с неуникальными ключами» – Multimap. Класс должен хранить набор пар (ключ, значение), причём ключи могут повторяться. Тип ключа и тип значения

должны быть параметрами шаблона. Написать несколько конструкторов, в том числе конструктор копирования. В классе реализовать методы для добавления нового элемента в отображение, удаления элемента, поиска по ключу и вывода на экран. Перегрузить операции для объединения (сложение), пересечения (умножения), индексирования и присваивания**.** 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.

Вариант 3/.

«Комплексное число» – Complex. Типы действительной и мнимой части должны быть параметрами шаблона. Класс должен содержать несколько конструкторов. В классе реализовать методы для сложения, вычитания, умножения и вывода на экран. Перегрузить операции для сложения, вычитания, умножения, присваивания и вывода на экран. Создать два вектора размерности n из комплексных координат. Передать их в функцию, которая выполняет сложение комплексных векторов.

Вариант 4.

«Очередь с двумя концами» – Deque. Написать несколько конструкторов, в том числе конструктор копирования. 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. В классе реализовать методы добавления элементов в очередь (в начала и в конец очереди) и удаление элементов из очереди (из начала и из конца очереди). Перегрузить операции сложения, вычитания, присваивания и вывода на экран. Создать массив объектов. Передавать объекты в функцию, которая удаляет из очереди первый (с головы очереди), третий, пятый и т. д. (нечётные) элементы.

Вариант 5.

«Одномерный массив» - Array размерности n**.** Разработать шаблонный класс для работы с одномерным массивом (размерность массива – также параметр шаблона). Написать несколько конструкторов, в том числе конструктор копирования. В классе реализовать методы для сортировки массива (пузырьковая сортировка), объединения двух массивов и для вывода на экран. Перегрузить операции индексирования, присваивания и вывода на экран для данного класса. Создать массив объектов и передавать объекты в функцию, которая выполняет объединение двух массивов и для полученного объекта-результата производит сортировку соответствующим методом.

Вариант 6.

«Очередь» – Queue. Написать несколько конструкторов, в том числе конструктор копирования. 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. В классе реализовать методы добавления элементов в очередь и удаление элементов из очереди. Перегрузить операции сложения, вычитания, присваивания и вывода на экран.

Создать массив объектов. Передавать объекты в функцию, которая удаляет из очереди первый (с головы очереди), третий, пятый и т. д. (нечётные) элементы.

Вариант 7.

«Упорядоченное бинарное дерево» – Tree. Тип хранимых в дереве ключей – параметр шаблона. Написать несколько конструкторов, в том числе конструктор копирования. В классе реализовать методы для вставки элемента в дерево, удаления элемента по ключу, поиска элемента по ключу, вывода на экран. Перегрузить операции сложения, вычитания, присваивания и вывода на экран. Сформировать дерево, вывести содержимое его узлов в порядке возрастания, найти узел по ключу, вывести содержимое листьев дерева (вершин, не имеющих потомков).

Вариант 8.

«Двусвязный список» – List. Написать несколько конструкторов, в том числе конструктор копирования. В классе реализовать методы для включения элемента в список, удаления и поиска элемента. Перегрузить операции сложения, вычитания, присваивания и вывода на экран. Создать два упорядоченных по возрастанию списка, слить их в один (также упорядоченный по возрастанию), построив новый список, и вывести результат на экран.

Вариант 9.

«Односвязный список» – List. Написать несколько конструкторов, в том числе конструктор копирования. В классе реализовать методы для включения элементов в список, удаления и поиска элемента. Перегрузить операции сложения, вычитания, присваивания и вывода на экран. Создать массив объектов. Передавать объекты в функцию, которая упорядочивает элементы списка по возрастанию и выводит результат.

Вариант 10.

«**Множество**» – **Set** (элементы уникальны). Написать несколько конструкторов, в том числе конструктор копирования. В классе реализовать методы для определения принадлежности заданного элемента множеству, пересечения, объединения, разности двух множеств. 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.

Вариант 11.

«**Мультимножество**» – **Multiset** (элементы могут повторяться). Написать несколько конструкторов, в том числе конструктор копирования. В классе реализовать методы для определения принадлежности заданного элемента мультимножеству, пересечения, объединения, разности двух мультимножеств. 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. ($A \cup B$) \ ($A \cap B$), и возвращает его в основную программу.

Вариант 12.

«Вектор» – Vector размерности n. Размерность вектора – также параметр шаблона. Класс должен содержать несколько конструкторов, в том числе конструктор копирования. В классе реализовать методы для вычисления модуля вектора, скалярного произведения, сложения, вычитания, умножения на константу. 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.

Вариант 13.

«Квадратная матрица» – Matrix. Размерности матрицы – также параметры шаблона. Класс должен содержать несколько конструкторов, в том числе конструктор копирования. В классе реализовать методы для сложения, вычитания, умножения матриц; вычисления нормы матрицы. 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. В основной программе вывести результат.

Вариант 14.

«Отображение с неуникальными ключами» – Multimap. Класс должен хранить набор пар (ключ, значение), причём ключи могут повторяться. Тип ключа и тип значения должны быть параметрами шаблона. Написать несколько конструкторов, в том числе конструктор копирования. В классе реализовать метод для поиска по ключу. Перегрузить операции для добавления нового элемента в отображение (сложение), удаления всех элементов по заданному 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. ($A \cup B$) \ ($A \cap B$), и возвращает его в основную программу.

Вариант 15.

«Одномерный массив» - Array размерности n. Разработать шаблонный класс для работы с одномерным массивом (размерность массива – также параметр шаблона). Написать несколько конструкторов, в том числе конструктор копирования. В классе реализовать методы для сортировки массива (сортировка слиянием), объединения двух массивов и для вывода на экран. Перегрузить операции индексирования, присваивания и вывода на экран для данного класса. Создать массив объектов и передавать объекты в функцию, которая выполняет объединение двух массивов и для полученного объекта-результата производит сортировку соответствующим методом.

Вариант 16.

«Многочлен» – Polynom степени *n*. Степень полинома также параметр шаблона. Написать несколько конструкторов, в том числе конструктор копирования. В классе реализовать методы для вычисления значения полинома; сложения, вычитания и умножения полиномов. Перегрузить операции сложения, вычитания, 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.

Вариант 17.

«Отображение» – Map. Класс должен хранить набор пар (ключ, значение), причём ключи уникальны. Тип ключа и тип значения должны быть параметрами шаблона. Написать несколько конструкторов, в том числе конструктор копирования. В классе реализовать методы для добавления нового элемента в отображение, удаления элемента, поиска по ключу и вывода на экран. 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.