

Реализовать соответствующий класс (без использования библиотеки STL). Память под члены-данные класса должна выделяться динамически (где это необходимо).

Вариант 1.

«Вектор целых чисел» – **Vector** размерности n . Класс должен содержать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для вычисления модуля вектора, скалярного произведения, сложения, вычитания, умножения на константу. Создать массив объектов. Написать функцию, которая для заданной пары векторов будет определять, являются ли они коллинеарными или ортогональными.

Вариант 2.

«Строка» – **String** длины n . Написать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для выполнения конкатенации строк, извлечения символа из заданной позиции, сравнения строк. Создать массив объектов и передать его в функцию, которая выполняет сортировку строк.

Вариант 3.

«Мультимножество целых чисел» – **Multiset** (элементы могут повторяться). Написать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для определения принадлежности заданного элемента мультимножеству, пересечения, объединения, разности двух мультимножеств. 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.

Вариант 4.

«Множество строк» – **Set**. Написать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для определения принадлежности заданного элемента множеству, пересечения, объединения, разности двух множеств. 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 5.

Разработать класс для «Массива строк». Написать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для поэлементной конкатенации двух массивов, упорядочения строк в лексикографическом порядке, слияния двух массивов с удалением повторяющихся строк, а также для вывода на экран всего массива и заданной строки. Создать массив объектов и передавать объекты в функцию, которая выполняет слияние объектов и для полученного объекта-результата производит лексикографическое упорядочения строк.

Вариант 6.

«**Множество целых чисел**» – **Set** (элементы уникальны). Написать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для определения принадлежности заданного элемента множеству, пересечения, объединения, разности двух множеств. 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 7.

«**Очередь целых чисел**» – **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. Включение элементов в очередь и их извлечение реализовать в виде методов класса. Создать массив объектов. Передавать объекты в функцию, которая удаляет из очереди первый (с головы очереди), третий, пятый и т. д. (нечётные) элементы.

Вариант 8.

«**Дробь**» – **Fraction**. Класс должен содержать несколько конструкторов. Реализовать методы для сложения, вычитания, умножения и деления дробей. Создать массив объектов и передать его в функцию, которая изменяет каждый элемент массива с четным индексом путем добавления следующего за ним элемента массива.

Вариант 9.

«**Многочлен с целыми коэффициентами**» – **Polynom** степени n . Написать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для вычисления значения полинома; сложения, вычитания и умножения полиномов. Создать массив объектов класса. Передать его в функцию, вычисляющую сумму полиномов массива и возвращающую полином-результат, который выводится на экран в основной программе.

Вариант 10.

«**Стек целых чисел**» – **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. Включение элементов в стек и их извлечение реализовать в виде методов класса. Создать массив объектов. Передавать объекты в функцию, которая удаляет из стека первый (сверху), третий, пятый и т. д. (нечётные) элементы.

Вариант 11.

«**Бинарное дерево целых чисел**» – **Tree**. Написать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для вставки элемента в дерево, удаления

элемента по ключу, поиска элемента по ключу, 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 12.

«Односвязный список целых чисел» – List. Написать несколько конструкторов, в том числе конструктор копирования. 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. Включение элемента в список, удаление и поиск элемента реализовать в виде методов класса. Создать массив объектов. Передавать объекты в функцию, которая упорядочивает элементы списка по возрастанию и выводит результат.

Вариант 13.

«Комплексное число» – Complex. Класс должен содержать несколько конструкторов. Реализовать методы для сложения, вычитания, умножения, деления и вывода на экран. Создать два вектора размерности n из комплексных координат. Передать их в функцию, которая выполняет сложение комплексных векторов.

Вариант 14.

«Отображение» – Map. Класс должен хранить набор пар (целочисленный ключ, целочисленное значение), причём все ключи уникальны. Написать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для добавления нового элемента в отображение, удаления элемента, поиска по ключу 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.

Вариант 15.

«Стек строк» – Stack. Написать несколько конструкторов, в том числе конструктор копирования. 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. Включение элементов в стек и их извлечение реализовать в виде методов класса. Создать массив объектов. Передавать объекты в функцию, которая удаляет из стека второй (сверху), четвёртый, шестой и т. д. (чётные) элементы.

Вариант 16.

«Квадратная матрица целых чисел» – Matrix. Класс должен содержать несколько конструкторов, в том числе конструктор копирования. Реализовать методы для сложения, вычитания, умножения матриц; вычисления нормы матрицы. 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 17.

«Двусвязный список целых чисел» – List. Написать несколько конструкторов, в том числе конструктор копирования. 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. Включение элемента в список, удаление и поиск элемента реализовать в виде методов класса. Создать два упорядоченных по возрастанию списка, слить их в один (также упорядоченный по возрастанию), построив новый список, и вывести результат на экран.