

Задача 1

Написать шаблон классов одномерных массивов. Предусмотреть конструктор копирования и оператор присваивания. Иметь в классе функцию-член сортировки массива. Написать тестирующую программу, в которой испытать шаблон на символьном массиве и массиве чисел с плавающей точкой.

Задача 2

Написать класс, содержащий строку с именем человека и 3 целых числа с датой его рождения (число, месяц и год). Предусмотреть в классе конструктор копирования и оператор присваивания и перегрузить операторы ввода и вывода. Определить функцию-оператор < для сравнения людей по именам. Написать тестирующую программу.

Задача 3

Написать класс, содержащий строку с именем человека и 3 целых числа с датой его рождения (число, месяц и год). Предусмотреть в классе конструктор копирования и оператор присваивания и перегрузить операторы ввода и вывода. Определить функцию-оператор < для сравнения людей по датам рождения. Написать тестирующую программу.

Задача 4

Написать класс, моделирующий матрицы 4×4 . Иметь в классе функцию, которая первой строкой делает строку с максимальным по модулю первым элементом.

Задача 5

Разработать класс для моделирования определителей 3-го порядка. Перегрузить оператор () для вычисления определителя и операторы + и * для сложения и умножения определителей. Написать тестирующую программу, в которой создать два определителя, вычислить их значения, сложить и перемножить определители, вычислить значение суммы определителей и значение произведения определителей.

Задача 6

Разработать класс для моделирования прямых на плоскости. Написать функцию-член этого класса, находящую точку пересечения двух прямых. Эта функция должна возвращать истину, если прямые пересекаются и ложь, если параллельны. Координаты точки пересечения должны возвращаться через аргументы функции, передаваемые в нее по ссылке. Написать тестирующую программу, в которой испытать данный класс.

Задача 7

Напишите класс, моделирующий плоскости в трехмерном пространстве. В классе напишите функцию-член, возвращающую расстояние от начала координат до плоскости. Напишите тестирующую программу.

Задача 8

Разработайте класс `Mat4` (матрица 4×4). Внутри этого класса объявите локальный вспомогательный класс `Mat3` для матриц 3-го порядка, в котором определите функцию-член для вычисления определителя 3-го порядка. В классе `Mat4` определите функцию для вычисления определителя путем разложения по 1-й строке. Напишите тестирующую программу.

Задача 9

Разработайте класс `Stack` (стек) из символов с реализацией в виде массива. В тестирующей программе используйте класс `Stack` для проверки правильности расстановки круглых скобок в тексте, расположенном в файле.

Задача 10

Напишите шаблон класса `Stack` (стек) для элементов произвольного типа. Используйте реализацию в виде массива. В тестирующей программе создайте два стека: из символов и из целых.

Задача 11

Напишите шаблон класса `Stack` (стек) для элементов произвольного типа. Используйте реализацию в виде односвязного списка. В тестирующей программе создайте стек из символов и из целых.

Задача 12

Разработайте класс `Pair` (пара: Имя - строка, Значение – число с плавающей точкой). Определите класс `Table` (таблица имен) в виде массива `Pair`. Напишите функцию-член `Add`, добавляющую в таблицу новое значение, если добавляемого имени не было в таблице, или заменяющее старое значение новым, если имя уже было в таблице. Предусмотрите генерацию исключения при переполнении таблицы. Напишите тестирующую программу.

Задача 13

Напишите программу создания предметного указателя для текста, содержащегося в файле. Ключевые слова располагаются в отдельном файле и разделены пробелами. Предметный указатель должен распечатываться в виде последовательности строк, в начале каждой из которых стоит ключевое слово, а далее – номера строк текста, на которых данное слово встречается.

Задача 14

Определите классы `TPoint` – точка на плоскости, `TOrt` – единичный вектор, задающий направление. Перегрузите для этих классов операторы вывода `<<` и ввода `>>`. На базе этих классов с использованием множественного наследования определите класс `TLine` – прямая на плоскости. Перегрузите операторы ввода и вывода для класса `TLine`, причем в классе `TLine` используйте операторы ввода и вывода из базовых классов. Напишите тестирующую программу.

Задача 15

Определите классы TPoint – точка на плоскости, TOrt – единичный вектор, задающий направление. На базе этих классов с использованием множественного наследования определите класс TLine – прямая на плоскости. Напишите функции, возвращающие коэффициенты A , B , C уравнения прямой $Ax+By+C = 0$. Нормируйте коэффициенты условием $A^2+B^2 = 1$. Напишите тестирующую программу.

Задача 16

Определите классы TPoint – точка на плоскости, TOrt – единичный вектор, задающий направление. На базе этих классов с использованием множественного наследования определите класс TLine – прямая на плоскости. Напишите функцию, нахождения срединного перпендикуляра к отрезку, заданному двумя точками. Напишите тестирующую программу.

Задача 17

Напишите класс Time (время) с членами: часы, минуты, секунды. Предусмотрите генерацию исключений при неверном задании времени. Переопределите оператор + для добавления к заданному моменту времени некоторого отрезка времени. Напишите тестирующую программу..

Задача 18

Разработайте класс Time (время), с представлением часы, минуты, секунды. В конструкторе класса Time предусмотрите генерацию исключения при неверной инициализации. В тестирующей программе организуйте обработку исключений.

Задача 19

В классе Time (время), с представлением часы, минуты, секунды, переопределите операторы $>$, $<$, $==$. Напишите тестирующую программу, в которой примените разработанный класс.

Задача 20

В классе Time (время), с представлением часы, минуты, секунды, переопределите оператор – для нахождения промежутка времени, прошедшего между двумя моментами времени. Предусмотрите в этом операторе генерацию исключения если из более раннего момента времени будет вычитаться более поздний момент. Напишите тестирующую программу.

Задача 21

Реализуйте класс Time (время суток), считая, что представлением времени суток является число секунд, прошедших от полуночи. Предусмотрите несколько конструкторов, в том числе конструктор, устанавливающий время по часам, минутам и секундам. Перегрузите оператор вывода $<<$, который должен выводить время в привычном виде часы:минуты:секунды. Напишите тестирующую программу.

Задача 22

Напишите класс `Polar` для работы с точками на плоскости в полярной системе координат. Определите функцию-оператор `+` для сложения точек, причем смысл этой операции должен состоять в нахождении точки, являющейся концом вектора, равного сумме векторов, соединяющих начало координат и складываемые точки.. Продемонстрируйте использование класса.

Задача 23

Напишите класс `Array` для работы с массивом элементов типа `double`. Предусмотрите в нем функцию-оператор `[]` для доступа к элементу массива по его номеру, в которой генерируйте исключение при выходе индекса за границы массива. Напишите функцию `main` для демонстрации работы с классом.

Задача 24

На базе класса `Circle` (окружность) разработайте класс `Wheel` (колесо), в котором базовая окружность является ободом колеса, имеется центральная окружность – ступица и член, задающий число спиц. Предусмотрите функцию, изображающую колесо. Спицы изображайте отрезками прямых. Напишите тестирующую программу.

Задача 25

На базе класса `Circle` (окружность) разработайте класс `Ring` (кольцо), в котором базовая окружность является внешней границей кольца, и имеется член, задающий ширину кольца. Предусмотрите в классе функцию для вычисления площади кольца. Напишите тестирующую программу.

Задача 26

Напишите класс, для моделирования географических объектов, а именно городов. В классе нужно хранить название города и его географические координаты – широту и долготу. Напишите функцию-член класса, вычисляющую расстояние между городами по поверхности Земли, то есть по дуге большого круга, получаемого сечением поверхности Земли, проходящей через города и центр Земли. Протестируйте класс.

Задача 27

Напишите класс `Geo`, моделирующий географические координаты на поверхности Земли (долгота, широта). Напишите класс `Town` (города), членами которого должно быть название города и его географические координаты как объект типа `Geo`. В главной функции создайте массив городов для некоторой страны и распечатайте их в порядке с севера на юг.

Задача 28

Напишите класс `Geo`, моделирующий географические координаты на поверхности Земли (долгота, широта).. Напишите класс `Town` (города), членами которого должно быть название города и его географические координаты как объект типа `Geo`. В главной функции создайте массив городов для некоторой страны и распечатайте их в порядке с востока на запад.

Задача 29

Напишите шаблон класса `Queue` (очередь) для элементов произвольного типа. Очередь реализуйте в виде массива. В тестирующей программе создайте очереди из символов и из целых чисел. Помещайте в очередь случайные целые неотрицательные числа, пока их сумма не превысит наперед заданного значения.

Задача 30

Разработайте класс, моделирующий рациональные числа как *несократимые* дроби, включающий числитель `Num` и знаменатель `Den`. Предусмотрите конструктор и переопределите в этом классе операторы сложения, вычитания, умножения и деления рациональных чисел. Функция, которая приводит дробь к несократимому виду, должна быть закрытым членом класса. Напишите главную функцию, иллюстрирующую работу с классом

Задача 31

Напишите шаблон функции, реализующей алгоритм быстрой сортировки массивов с элементами произвольного типа. В тестирующей программе создайте массивы из символов и из целых чисел и отсортируйте их.

Задача 32

Реализуйте класс `Date` (дата), считая, что представлением даты является число дней, прошедших после 1.1.1970. Предусмотрите в классе функцию-член, которая возвращает через свои аргументы, имеющие тип ссылок на `int`, значения дня, месяца и года. Напишите тестирующую программу.

Задача 33

При входе в метро некоторые пассажиры оплачивают проезд, а некоторые проходят бесплатно по удостоверения. Разработайте класс, для учета пассажиров и заплаченных ими денег. В классе иметь два члена: количество вошедших на станцию метро пассажиров и сумма заплаченных ими денег. Предусмотрите две функции-оператора `+` для добавления целого (увеличивает число пассажиров и не увеличивает выручку) и для добавления числа с плавающей точкой (увеличивает сумму денег и количество пассажиров на единицу). Напишите метод для показа числа пассажиров и общей выручки. Протестируйте программу, промоделировав поток пассажиров через две станции метро.

Задача 34

Разработать класс, моделирующий комплексные числа. Включить в него функции:

$$e^z = e^{x+iy} = \cos x + i \sin y; \quad \sin z = \frac{e^z - e^{-z}}{2i}; \quad \cos z = \frac{e^z + e^{-z}}{2}.$$

В тестирующей программе убедитесь в справедливости основного тригонометрического тождества для комплексных чисел.

Задача 35

Разработайте класс, моделирующий многочлены n – го порядка от одной переменной. В классе должен быть конструктор копирования и оператор присваивания. Перегрузите оператор вызова функции $()$ с одним аргументом (`double x`) для вычисления значения многочлена при заданном x . Вычисление многочлена производить методом Горнера. Напишите тестирующую программу.

Задача 36

Разработайте класс, моделирующий многочлены n – го порядка от одной переменной. В классе должен быть конструктор копирования и оператор присваивания. Перегрузите оператор \sim для вычисления первой производной многочлена. Напишите тестирующую программу, в которой задайте некоторый многочлен и вычислите его производную.

Задача 37

Разработайте класс, моделирующий многочлены n – го порядка от одной переменной. В классе должен быть конструктор копирования и оператор присваивания. Перегрузите оператор $!$ для вычисления неопределенного интеграла для данного многочлена. Произвольную постоянную, появляющуюся при интегрировании, считайте, для определенности, равной 1. Напишите тестирующую программу, в которой задайте некоторый многочлен и вычислите его интеграл.

Задача 38

Разработайте класс, моделирующий прямоугольные матрицы размера $n * m$, где n – число строк, m – число столбцов. Элементы матрицы представьте в виде одномерного массива, в котором идут подряд строки матрицы, начиная с первой строки. Предусмотрите конструктор, переопределите функцию оператор $()$ с двумя аргументами (`int i, int j`), которая возвращала бы ссылку на элемент i – й строки, j – го столбца матрицы. Напишите тестирующую программу.

Задача 39

Напишите класс, моделирующий матрицы порядка $n*m$. Матрицу представьте как одномерный массив, в котором построчно расположены элементы матрицы. Напишите функцию-член класса, которая транспонирует матрицу. Протестируйте класс.

Задача 40

Объявите класс, моделирующий многомерные вектора, членами которого пусть являются указатель на массив элементов и размер вектора. Объявите класс матриц, членами которого пусть будут два целых числа – размеры матрицы и указатель на массив, в котором последовательно расположены строки матрицы. Объявите класс векторов другом класса матриц. Напишите функции умножения матрицы на вектор и вектора на матрицу. Протестируйте данные классы.

Задача 41

Напишите класс для работы с n -мерными векторами. Кроме размерности n сделайте членом класса указатель на массив элементов вектора. Напишите конструктор копирования и оператор присваивания. Перегрузите оператор вызова функции $()$ для вычисления угла между двумя векторами. Напишите тестирующую программу.

Задача 42

Разработайте класс для моделирования прямоугольников, которые могут иметь наклонное положение относительно осей координат, как производный от класса `Point` (точка на плоскости). Базовую точку считайте центром прямоугольника. Членами класса могут быть длины сторон и угол между одной из сторон прямоугольника и осью x . Определите функцию с аргументом – массивом из 4-х `Point`, которая заполняет аргумент-массив точками – вершинами прямоугольника. Предусмотрите функцию поворота прямоугольника на некоторый угол вокруг своего центра. Протестируйте разработанный класс.

Задача 43

На базе класса `Point` (точка на плоскости) определите класс `Rect` (прямоугольник со сторонами, параллельными осям координат). Базовую точку считайте центром прямоугольника, членами класса считайте длину и ширину прямоугольника. Перегрузите в классе `Rect` оператор $+$, который должен возвращать площадь фигуры, полученной наложением двух прямоугольников. Напишите тестирующую программу.

Задача 44

На базе класса `Point` (точка на плоскости) определите класс `Rect` (прямоугольник со сторонами, параллельными осям координат). Базовую точку считайте центром прямоугольника, членами класса считайте длину и ширину прямоугольника. Перегрузите в классе `Rect` оператор $/$, который должен возвращать площадь фигуры, полученной пересечением двух прямоугольников. Напишите тестирующую программу.

Задача 45

На базе класса `Point` (точка на плоскости) определите класс `Rect` (прямоугольник со сторонами, параллельными осям координат). Базовую точку считайте центром прямоугольника, членами класса считайте длину и ширину прямоугольника. Перегрузите в классе `Rect` оператор $-$, который должен возвращать площадь фигуры, полученной вычитанием из первого прямоугольника второго. Напишите тестирующую программу.

Задача 46

Используя класс `Point` (точка на плоскости) как базовый определите класс прямоугольников со сторонами, параллельными осям координат. Базовую точку считать одной из вершин прямоугольника, прямоугольник определять еще одной точкой, расположенной по диагонали от первой. Предусмотрите в классе функцию, которая делает базовую точку левой нижней, а другую точку – правой верхней. Переопределите функцию-оператор $+$ для получения минимального прямоугольника, который включает два заданных прямоугольника. Напишите тестирующую программу.

Задача 47

Определите класс `Vec4` как вектор из 4-х `float`. Определите оператор `[]` для доступа к элементам вектора. Определите операторы `+` для сложения векторов и `*` для умножения вектора на число. Определите оператор `~` для перестановки элементов вектора в обратном порядке и операторы ввода `>>` и вывода `<<`. Напишите тестирующую программу.

Задача 48

Напишите класс `Vec4` – массив из 4-х `float`. Определите класс `Mat4` (матрица 4*4) как массив из 4-х `Vec4`. В классах предусмотрите необходимые конструкторы. Определите оператор `*` для умножения матрицы на вектор и вектора на матрицу. Напишите тестирующую программу.

Задача 49

Напишите класс для работы со строками символов. Переопределите в классе оператор `+` для конкатенации строк. Напишите тестирующую программу.

Задача 50

Напишите класс для работы со строками символов. Переопределите оператор вызова функции (`int k, int p`) для выделения подстроки из `k` символов, начиная с символа `p`. Протестируйте класс.

Задача 51

Разработайте класс, моделирующий вектора в трехмерном эвклидовом пространстве с возможностью выполнения операций сложения, вычитания, умножения на скаляр, скалярного и векторного произведений. Напишите главную функцию, иллюстрирующую работу класса.

Задача 52

Разработайте класс `Integral`, для моделирования определенного интеграла. Членами класса должны быть указатель на подынтегральную функцию, пределы интегрирования и значение интеграла. Интеграл вычисляйте в конструкторе каким-либо методом (прямоугольников, трапеций и т.п.). Предусмотрите функцию-член вычисления интеграла на новом отрезке. Протестируйте класс.

Задача 53

Определите абстрактный класс `Figure` (фигура на плоскости), включающий координаты `x`, `y` характерной точки фигуры. Предусмотрите в классе `Figure` функции-члены `show`, `nshow`, `move`. Определите на базе класса `Figure` конкретный класс `Square` (квадрат). Центр квадрата задается базовой точкой, размер квадрата задается длиной `a` стороны, ориентация – углом поворота `angle` вокруг центра. Включите функцию-член `rotate(double al)`, поворачивающую квадрат вокруг базовой точки на угол `al`. Напишите тестирующую программу.

Задача 54

Определите абстрактный класс `Figure` (фигура на плоскости), включающий координаты x , y характерной точки фигуры. Предусмотрите в классе `Figure` функции-члены `show`, `noshow`, `move`. Определите на базе класса `Figure` конкретный класс `Triangle` (треугольник). Пусть базовая точка задает координаты одной вершины треугольника, кроме этого пусть в класс `Triangle` входят длины трех сторон a , b , c и угол между стороной a и осью x . Напишите тестирующую программу.

Задача 55

Написать класс `Point` (точка на плоскости) и производный от него класс `Circle` (окружность). В классе `Circle` написать функцию-член `int intersect_x(double& x1, double& x2)`, которая возвращает количество точек пересечения окружности с осью x и заносит координаты точек пересечения в $x1$, $x2$. Написать тестирующую программу.

Задача 56

Объявите класс `Point` (точка на плоскости). На его базе определите абстрактный класс `Figure` (фигура), предусмотрите в нем функции члены `show`, `noshow`, `move`. Определите на базе класса `Figure` конкретный класс `Circle` (окружность). Определите в классе `Circle` функцию-оператор `+` сложения двух окружностей, которая возвращает окружность минимального радиуса, внутри которой находятся две исходные окружности. Напишите тестирующую программу.

Задача 57

Разработайте класс `Pair` (пара: Имя - строка, Значение – число с плавающей точкой). Определите класс `Table` (таблица имен) в виде массива `Pair`. Напишите функцию-член `Add`, добавляющую в таблицу новое значение, если добавляемого имени не было в таблице, или заменяющее старое значение новым, если имя уже было в таблице. Предусмотрите генерацию исключения при переполнении таблицы. Напишите тестирующую программу.

Задача 58

Напишите шаблон функции для сортировки массивов с элементами произвольного типа. Используйте алгоритм пузырьковой сортировки. Напишите тестирующую программу, в которой создайте массивы из целых чисел и из чисел с плавающей точкой и отсортируйте их. Для вывода массива с элементами произвольного типа напишите шаблон функции.

Задача 59

Напишите класс, моделирующий квадратные трехчлены. Перегрузите оператор `+` для сложения трехчленов. В составе класса должна быть функция, вычисления корней `int root(double& x1, double& x2)`, которая возвращает число корней квадратного трехчлена, и записывает их значения в аргументы $x1$, $x2$. Напишите тестирующую программу, в которой создайте два трехчлена, вычислите их корни, найдите сумму трехчленов и корни суммы.

Задача 60

Напишите класс упорядоченных массивов произвольного размера. В классе должна быть закрытая функция-член сортировки массива, которую следует вызывать в конструкторе при создании массива.. Перегрузите функцию-оператор + для создания нового упорядоченного массива по двум существующим, в которой новый упорядоченный массив создавайте слиянием двух исходных без использования сортировки. Напишите тестирующую программу.

Задача 61

Организируйте помещение вводимых символов в список, пока не будет достигнут конец входного потока. Затем переместите буквы из исходного списка в отдельный список букв, цифры – в список цифр, а остальные символы – в третий список. Выведите созданные списки.

Задача 62

Организируйте помещение вводимых чисел в список, так чтобы получился список, упорядоченный по возрастанию.

Задача 63

Добавьте в класс односвязных линейных списков метод `bool IsOrdered()`, проверяющий, является ли список упорядоченным по возрастанию.

Задача 64

Напишите функцию `List& merge(List& L1, List& L2)`, добавляющую к упорядоченному односвязному линейному списку L1 также упорядоченный односвязный линейный список L2 и возвращающую ссылку на результирующий список.

Задача 65

Добавьте в класс односвязных линейных списков конструктор копирования и оператор присваивания и напишите программу с их использованием.

Задача 66

Добавьте в класс односвязных линейных списков функцию `Unique`, которая удаляет из списка повторяющиеся элементы, в результате чего все элементы списка должны быть различными. Протестируйте данную функцию.

Задача 67

Добавьте в класс односвязных линейных списков метод `Swap(int i, int j)`, меняющую информацию звеньев списка с номерами *i* и *j*.

Задача 68

Добавьте в класс односвязных линейных списков метод `int Search(T x)` для поиска значения `x` в списке. Метод должен возвращать номер узла со значением `x` или `-1`, если `x` не содержится в списке.

Задача 69

Добавьте в класс односвязных линейных списков метод `bool Inser(T x)` вставляющий `x` в список, если такого значения в списке не было. Метод должен возвращать истину, если `x` добавлено в список и ложь, если `x` уже был в списке.

Задача 70

Добавьте в класс односвязных линейных списков метод `Sort` сортировки списка, например, методом пузырька. При сортировке сами звенья не переставляйте, переставляйте только информацию звеньев.